

LA36

LA36 TERM DL11 & KL11
CZLACFO

AH 8899F MI

NOV 19 1964

COPYRIGHT 1964

digital

FICHE 1 OF 1

MADE IN USA

Microfiche grid containing multiple frames of data, including headers and tables. The text is extremely faint and illegible due to the low resolution of the scan.

.REM_

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

IDENTIFICATION

PRODUCT CODE: AC-8898F-MC
PRODUCT NAME: CZLACFO LA36 TERM (DL11 & KL11)
DATE: AUGUST 1978
MAINTAINER: DIAGNOSTIC GROUP

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1974,1979 BY DIGITAL EQUIPMENT CORPORATION

HISTORY

- 1.0 DECO CZLAC-F-0
 - 1.1.0 CLOSED PROBLEM REPORT AA3318
 - 1.1.1 IT WAS REPORTED THAT A DL11-A OPERATED AT 110 BAUD CAUSED FAILURE IN THE AREAD ROUTINE BECAUSE THE 200 MSEC. DELAY IS NOT OF SUFFICIENT DURATION TO ALLOW SETTING OF THE RECEIVER REGISTER STATUS "DONE" BIT THROUGH THE MAINTENANCE BIT FACILITY. THE TIME DELAY WAS INCREASED FROM 200 TO 600 MSEC.
 - 1.2.0 CLOSED PROBLEM REPORT AA3643
 - 1.2.1 TESTS 56,57,60,61,62,63,64,65,AND 66 DO NOT RUN PROPERLY WHEN RUN ON AN LSI-11. THIS PROBLEM WAS RESOLVED BY CHANGING THE BRANCH AFTER THE CHAIN COMMAND TO GO BACK TO TEST FOR THE LSI-11 SWITCH IN ORDER TO EFFECT THE APPROPRIATE ACTION DURING EACH TEST.
 - 1.2.2 SECOND TIME-OUT IN TEST 64 ALLOWS EXCESSIVE WAIT FOR OPERATOR RESPONSE. THE TIME DELAY WAS REDUCED FROM "177777" TO "600".
 - 1.2.3 COMMON ROUTINE TYPE DOES NOT SAVE THE CONTENTS OF RO RESULTING IN THE LOSS OF THIS INFORMATION AND CONSEQUENT FAILURE. INSTRUCTIONS WERE INCLUDED TO SAVE THE CONTENTS OF RO ON ENTRY INTO THE ROUTINE AND TO RESTORE THEM UPON EXIT.
 - 1.2.4 LOSS OF STACK CONTENTS FOR NON-LSI-11 COMPUTERS DUE TO INCORRECT SEQUENCE OF INSTRUCTIONS IN TEST 65 WAS ALSO REPORTED IN PROBLEM REPORT AA3803. REFER TO 1.3.1.
 - 1.3.0 CLOSED PROBLEM REPORT AA3803
 - 1.3.1 TESTING OF NON-LSI-11 COMPUTERS RESULTS IN THE PROGRAM HANGING UP BECAUSE THE STACK GETS POPPED AWAY IN TEST 65. THE BRANCH AFTER THE TEST FOR THE LSI-11 SWITCH IN TEST 65 SHOULD GO TO THE CHAIN COMMAND FOR PROPER EXIT FROM THE TEST FOR NON-LSI-11 MACHINES. THIS CHANGE SUPERSEDES THE CHANGE RELEASED IN DEPO MD-11-DZLAC-D-1.
 - 1.4.0 PROBLEM REPORT AA2390
 - 1.4.1 PROCESSORS WITHOUT HARDWARE SWITCH REGISTERS ARE NOT ABLE TO LOOP ON ERROR WITH THIS PROGRAM. INSTALLED ^G FUNCTION TO PERMIT CHANGING SWITCH REGISTER SETTING DURING DIAGNOSTIC TESTING.

45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT AND ASSIGNMENTS
2.2	STORAGE
2.3	PRELIMINARY PROGRAMS
2.4	ADDITIONAL PROGRAMS
3.0	LOADING PROCEDURE AND INITIALIZATION
4.0	STARTING PROCEDURE
4.1	STARTING ADDRESSES
4.2	SWITCH REGISTER CONTROL WITH I/O TESTS
4.3	SWITCH REGISTER CONTROL WITHOUT I/O TESTS
4.4	KEYBOARD CONTROL WITH I/O TESTS
4.5	KEYBOARD CONTROL WITHOUT I/O TESTS
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
6.4	STANDARD I/O TESTS

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204

1.0 ABSTRACT

THIS DIAGNOSTIC IS DIVIDED INTO THREE 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.

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 AND ASSIGNMENTS

THE DIAGNOSTIC IS WRITTEN TO RUN ON ALL MODELS OF THE PDP-11 COMPUTER WITH EITHER A KL11 OR DL11 CONSOLE TERMINAL INTERFACE. THE DIAGNOSTIC IS PRESET TO TEST UP TO 16 ADDITIONAL TERMINALS (ON DL11'S) ASSIGNED BETWEEN ADDRESSES 776500 AND 776676. THIS PRESET QUANTITY (16) AND PRESET ADDRESS (776500) CAN BE CHANGED BY DEPOSITING THE QUANTITY IN DLNR AND THE STARTING ADDRESS IN DLADR. FOR EXAMPLE, TO ALLOW FOR UP TO 31 ADDITIONAL TERMINALS, THE ADDRESS 775610 COULD BE PLACED INTO DLADR AND THE OCTAL EQUIVALENT OF 31, I.E., (37) WOULD BE PLACED INTO DLNR. THE NUMBER OF ADDITIONAL DL11'S ACTUALLY TESTED WILL BE ADJUSTED AUTOMATICALLY DOWNWARD BASED UPON THE FIRST DL11 ADDRESS (WITHIN THE IMPLIED RANGE) FOUND TO BE UNRESPONSIVE. THUS IF THERE IS NO DL11 PRESENT TO MATCH THE ADDRESS IN DLADR ONLY THE CONSOLE TERMINAL WILL BE TESTED. THEREFORE, ALL DL11'S IN EXCESS OF THE CONSOLE TERMINAL MUST HAVE CONTIGUOUS ADDRESS ASSIGNMENTS WITH THE LOWEST ADDRESS CORRESPONDING TO THE VALUE IN DLADR.

THE CONSOLE TERMINAL (ASSIGNED STANDARD) CAN BE REASSIGNED BY PLACING THE ADDRESS OF ITS RECEIVER STATUS REGISTER INTO CONADD AND ITS RECEIVER INTERRUPT VECTOR INTO CONVEC. THIS REASSIGNMENT CAN BE MADE TO A TERMINAL WITHIN THE SET OF TERMINALS IMPLIED BY DLNR AND DLADR WITHOUT ADVERSE EFFECT. NOTE THAT A TERMINAL WITH A SLOWER SPEED (IF ANY) WILL DETERMINE THE SPEED AT WHICH ALL OF THE TERMINALS ARE TESTED. SUCH A TERMINAL SHOULD GENERALLY BE EXCLUDED FROM THE TEST, OR TESTED SEPARATELY. (REFER TO THE SYMBOL DEFINITIONS IN THE LISTING FOR THE ABOVE MENTIONED LOCATIONS.)

2.2 STORAGE

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

205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

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.

2.4 ADDITIONAL PROGRAMS

THIS DIAGNOSTIC IS FOR VERIFICATION OF BASIC TERMINAL FUNCTIONS ONLY. IF THE TERMINALS UNDER TEST HAVE HARDWARE OPTIONS INSTALLED RUN DIAGNOSTIC MAINDEC-11-DZLAF-A, THE LA36 TERMINAL OPTIONS TEST.

3.0 LOADING PROCEDURE AND INITIALIZATION

LOAD THE LA36 DIAGNOSTIC PROGRAM TAPE FOLLOWING NORMAL PROCEDURES. BEFORE STARTING THE PROGRAM, 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-11/05 WITH CORE MEMORY. REFER TO SECTION 2.1 FOR NON-STANDARD TERMINAL ADDRESSES AND FOR TESTING MULTIPLE DL11 INTERFACES.

IF A HARDWARE SWITCH REGISTER DOES NOT EXIST, THE PROGRAM WILL USE THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES. WHEN THE PROGRAM STARTS UP, IT WILL ASK FOR A VALUE TO BE ENTERED BEFORE THE PROGRAM WILL PROCEED.

IF THE CPU IS AN LSI-11, 11/03 BE SURE TO SET SWITCH REGISTER BIT 9 TO A 1. SPECIAL TESTS ARE RUN ON THE DLV11 INTERFACE.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESSES

- 200(8) = RUN WITH SWITCH REGISTER CONTROL
 - PERFORM CONSOLE TERMINAL I/O TESTS.
- 204(8) = RUN WITH SWITCH REGISTER CONTROL
 - SKIP CONSOLE TERMINAL I/O TESTS.
- 210(8) = RUN WITH KEYBOARD CONTROL
 - PERFORM CONSOLE TERMINAL I/O TESTS.
- 214(8) = RUN WITH KEYBOARD CONTROL
 - SKIP CONSOLE TERMINAL I/O TESTS.

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

4.2 SWITCH REGISTER CONTROL WITH I/O TESTS

- A. SET THE SWITCH REGISTER TO 200(8) AND PRESS THE LOAD ADDRESS SWITCH.
- B. SET SWITCH REGISTER BIT 9 TO A 1 IF THE PROCESSOR IS AN LSI-11, 11/03. REFER TO SECTION 5.1.5.
- C. SET THE SWITCH REGISTER BITS 7-0 EQUAL TO THE PAPER WIDTH IN TERMS OF THE NUMBER OF COLUMNS (OCTAL). REFER TO SECTION 5.1.8.
- D. SET THE SWITCH REGISTER BIT 8 EQUAL TO 1 OR 0 AND PRESS THE START SWITCH. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER OF DL11'S BEING TESTED. REFER TO SECTION 5.1.6.
- E. IF BIT 8 WERE ZERO WHEN STARTING, THE PRINTER TESTS ARE EXECUTED SEQUENTIALLY, AFTER THE ENTIRE SERIES OF I/O TESTS ARE EXECUTED.
- F. IF BIT 8 WAS SET WHEN THE START SWITCH WAS PRESSED, THE ENTIRE SERIES OF I/O TESTS WILL BE EXECUTED AND THE CPU WILL HALT AT LOCATION SELHLT. THE PROGRAM WILL THEN BE WAITING FOR CONTROL VIA THE SWITCH REGISTER.

4.3 SWITCH REGISTER CONTROL - WITHOUT I/O TESTS

SAME AS SECTION 4.2 EXCEPT IN STEP A, SET THE SWITCH REGISTER TO 204(8).

293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335

4.4 KEYBOARD CONTROL - WITH I/O TESTS

- A. SET THE SWITCH REGISTER TO 210(8) AND PRESS THE LOAD ADDRESS SWITCH.
- B. SET THE SWITCH REGISTER BITS 7-0 EQUAL TO THE PAPER WIDTH IN TERMS OF THE NUMBER OF COLUMNS (OCTAL). REFER TO SECTION 5.1.8.
- C. SET SWITCH REGISTER BIT 9 TO A 1 IF THE PROCESSOR IS AN LSI-11, 11/03. REFER TO SECTION 5.1.5.
- D. SET SWITCH 8 AND PRESS THE START SWITCH. A MESSAGE WILL BE PRINTED INDICATING THE NUMBER OF DL11'S BEING TESTED. REFER TO SECTION 5.1.6.
- E. IF BIT 8 WAS ZERO WHEN STARTING, THE PRINTER TESTS ARE EXECUTED SEQUENTIALLY AFTER THE ENTIRE SERIES OF I/O TESTS ARE EXECUTED.
- F. IF BIT 8 WERE SET WHEN THE START SWITCH WAS PRESSED, THE ENTIRE SERIES OF I/O TESTS WILL BE EXECUTED FOLLOWED BY THE SELECT TEST MESSAGE. THE PROGRAM WILL THEN BE WAITING FOR A TEST SELECTION VIA ANY TERMINAL KEYBOARD. REFER TO SECTION 5.2.

4.5 KEYBOARD CONTROL - WITHOUT I/O TESTS

SAME AS SECTION 4.4 EXCEPT IN STEP A, SET THE SWITCH REGISTER TO 214 (8).

5.0 OPERATING PROCEDURE

THE PROGRAM CAN BE CONTROLLED IN EITHER OF TWO METHODS: BY THE CONSOLE SWITCH REGISTER OR FROM THE KEYBOARD OF THE TERMINAL(S) UNDER TEST.

336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389

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
14	1(UP) = CONTINUE ON ERROR 0(DOWN) = HALT ON ERROR
13	1(UP) = DRIVE ONLY CONSOLE TERMINAL 0(DOWN) = DRIVE ALL TERMINALS
11	1(UP) = LOOP ON INDIVIDUAL TEST 0(DOWN) = NORMAL TEST SEQUENCE
9	1(UP) = CPU TYPE IS AN LSI-11, 11/03 0(DOWN) = ALL OTHER PDP-11'S
8	1(UP) = RUN TEST ONCE AND HALT 0(DOWN) = LOOP ON TEST SEQUENCE
5-0	TEST NUMBER SELECTION
7-0	NUMBER OF COLUMNS AT START-UP

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. DURING THE HALT, ANY OF THE CONTROL SWITCHES MAY BE CHANGED OR SET AS DESIRED.

5.1.2 SWITCH 14

PLACING SWITCH 14 IN THE UP POSITION WILL CAUSE THE PROGRAM TO CONTINUE ON ERRORS DURING ANY OF THE I/O TESTS ONLY. WITH SWITCH 14 DOWN, THE PROGRAM WILL HALT (AT ERRHLT) ON ANY ERRORS DURING THE I-O TESTS WITH THE LOCATION OF THE ERROR IN RO. PRESSING CONTINUE WILL CAUSE THE PROGRAM TO CONTINUE IF SWITCH 14 IS DOWN. WITH SWITCH 14 UP, PRESSING CONTINUE WILL CAUSE THE PROGRAM TO LOOP ON THE ERROR.

390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443

NOTE

ERROR HALTS CAN OCCUR ONLY DURING THE I/O TESTS. THE TERMINAL IS CONNECTED TO A SERIAL LINE AND THERE IS NO ERROR INFORMATION RETURNED TO THE PROGRAM FROM THE TERMINAL. THEREFORE THE PROGRAM CANNOT REPORT ERRORS OCCURRING IN THE TERMINAL. ERRORS DETECTED DURING THE INTERFACE TESTS WILL RESULT IN HALTS AS DESCRIBED ABOVE.

5.1.3 SWITCH 13

PLACING SWITCH 13 IN THE DOWN POSITION WILL CAUSE THE DRIVING OF ALL MULTIPLE TERMINALS DURING THE PRINTER TESTS ONLY. IF SWITCH 13 IS UP, ONLY THE CONSOLE TERMINAL IS DRIVEN.

** NOTE: SWITCH 13 SHOULD ONLY BE CHANGED WHEN THE PROGRAM IS WAITING FOR A TEST SELECTION.

5.1.4 SWITCH 11

PLACING SWITCH 11 UP AT ANY TIME WILL CAUSE THE PROGRAM TO LOOP ON THE CURRENT TEST AS LONG AS SWITCH 11 REMAINS UP. REPLACING SWITCH 11 DOWN WILL CAUSE THE PROGRAM TO RESUME NORMAL OPERATION AT THE COMPLETION OF THE TEST.

5.1.5 SWITCH 9

PLACING SWITCH 9 UP AT THE START OF THE TEST WILL CAUSE AN AUTOMATIC CHANGE IN THE DELAY TIMING, AND THE EXECUTION OF SPECIAL DLV11 I/O TESTS. THE DLV11 HAS NO MAINTENANCE MODE AND WILL CAUSE THE PROGRAM TO HANG IF TESTED AS A DL11.

5.1.6 SWITCH 8

WITH SWITCH 8 IN THE DOWN POSITION THE PROGRAM WILL CONTINUE TO LOOP THROUGH THE PRESENT TEST SEQUENCE. PLACING SWITCH 8 UP WILL CAUSE THE PROGRAM TO HALT (AT SELHLT) AT THE COMPLETION OF THE CURRENT TEST. AFTER THE HALT, SET THE CONTROL SWITCHES AS DESIRED AND SET SWITCHES 5 TO 0 TO THE NEXT DESIRED TEST NUMBER, AND THEN PRESS CONTINUE TO START THE TEST.

444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470

WHEN STARTING THE DIAGNOSTIC THE OPERATOR CAN SELECT A SPECIFIC TEST RATHER THAN AUTOMATICALLY STARTING THE PRINTING TEST SEQUENCE BY SETTING SWITCH 8 UP BEFORE STARTING THE DIAGNOSTIC. UPON COMPLETION OF THE I/O TEST SEQUENCE (IF BEING RUN) THE PROGRAM WILL EITHER HALT AT SELHLT WAITING FOR A TEST SELECTION VIA THE SWITCH REGISTER OR PRINT THE SELECT TEST MESSAGE AND WAIT FOR A TEST SELECTION FROM ANY KEYBOARD. REFER TO SECTION 4 FOR FURTHER INFORMATION.

5.1.7 SWITCHES 5 TO 0

SWITCHES 5 TO 0 ARE USED TO SELECT SPECIFIC TESTS WHEN UNDER SWITCH REGISTER CONTROL. TEST NUMBERS ARE ALWAYS IN OCTAL.

5.1.8 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. THUS, FOR NORMAL OPERATION LEAVE SWITCHES 7 TO 0 DOWN TO TEST THE FULL 132(10) COLUMNS.

471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526

5.2 KEYBOARD CONTROL

THE PROGRAM WILL BE UNDER KEYBOARD CONTROL WHENEVER THE DIAGNOSTIC IS STARTED AT LOCATION 210 OR 214. SWITCHES ON THE CONSOLE SWITCH REGISTER WILL HAVE NO EFFECT WHEN UNDER TERMINAL CONTROL EXCEPT FOR SWITCH 15. THE I/O TESTS CANNOT BE SELECTED WHEN UNDER KEYBOARD CONTROL.

TO STOP A TEST AT ANY TIME, TYPE THE 'RUBOUT' OR 'DELETE' KFY ON ANY KEYBOARD. ANY TERMINAL MAY STOP THE TEST AND SELECT THE NEXT TEST IF SWITCH 13 IS DOWN. WHEN A TEST IS STOPPED BY TYPING A 'RUBOUT' OR 'DELETE', THE TEST WILL TERMINATE AND THE FOLLOWING MESSAGE WILL BE TYPED:

SELECT TEST NUMBER

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 OR LOWER CASE, BUT THE TEST NUMBER MUST ALWAYS BE A 2 DIGIT OCTAL NUMBER. THE TEST NUMBER AND TERMINATOR ARE ECHOED BY THE PROGRAM, THUS EACH CHARACTER WILL BE PRINTED TWICE IF THE TERMINAL IS IN HALF DUPLEX. 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 @ LOCATION 200 OR 204. CONSOLE SWITCH REG. VALUES WILL BE ENTERED FROM THE TTY AT START UP TIME WITH THE OLD VALUE INDICATED. TO CHANGE A VALUE TYPE ^G. IF TEST WAS STARTED @ LOC 200, AT THE END OF A PRINT TEST THE NEW VALUE CAN BE ENTERED. IF TEST WAS STARTED AT LOC 204, DATA CAN BE CHANGED BEFORE TEST FINISHES. ^U ALLOWS INCORRECT DATA ENTRIES TO BE DELETED AND A NEW ENTRY STARTED.

580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634

EXAMPLE :

...	AAA	AAA
...	BBB	BBB
...	CCC	CCC
...	DDD	DDD
...	EEE	EEE
...	FFF	FFF
...	GGG	GGG
...	HHH	HHH
...	III	III
...	JJJ	JJJ
...	KKK	KKK
...	LLL	LLL
...	MMM	MMM
...	NNN	NNN
...	OOO	OOO
...	PPP	PPP
...	QQQ	QQQ
...	RRR	RRR
...	SSS	SSS
...	TTT	TTT
...	UUU	UUU
...	VVV	VVV
...	WWW	WWW
...	XXX	XXX
...	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.

635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687

EXAMPLE:

001	SOH	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:

OXOXOXOXOXOXOXOXOXOXOXOXOXOXOX

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 DIAGONAL 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
          X

```

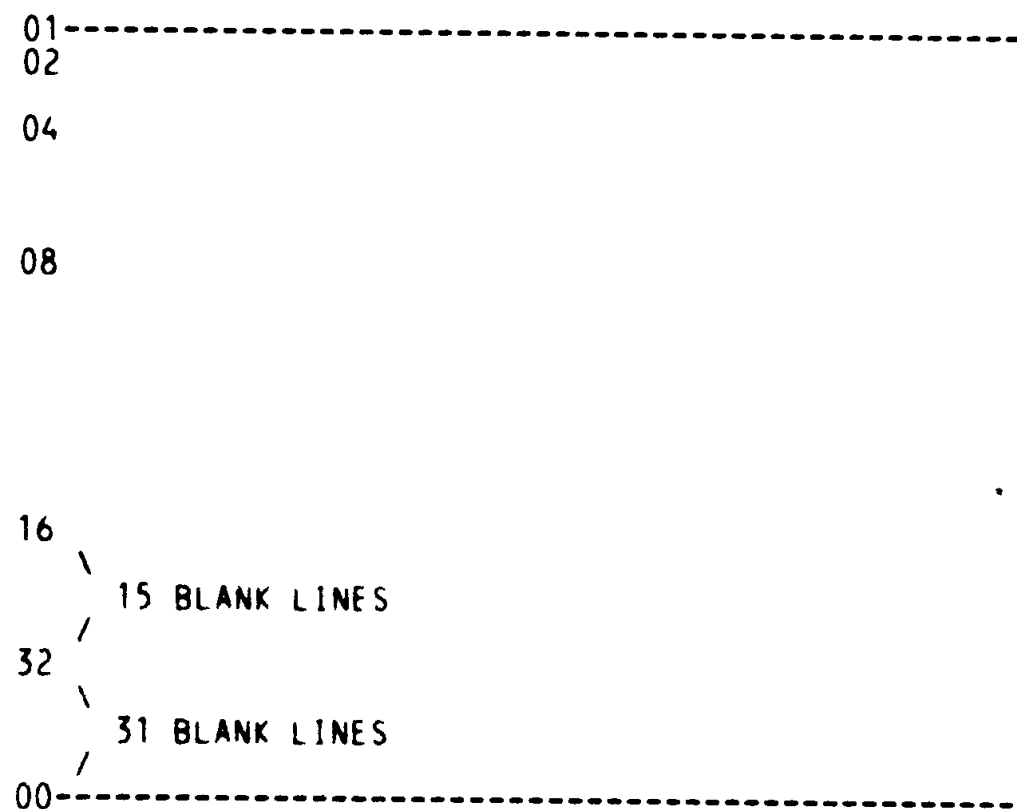
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731

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 BETWEEN THE TWO DASHED LINES, I.E., 63(10) 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:



732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769

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 MISPRINTS OR MISSING CHARACTERS DURING THE FIRST 16(10) CHARACTERS OF THE MIDDLE REFERENCE LINE. ALSO, ANY PAPER FEED PROBLEMS WILL CAUSE MISALIGNMENT OF THE SLASHES FORMING THE DIAGONAL LINE.

770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815

EXAMPLE:

0000000000000000000000000000000022

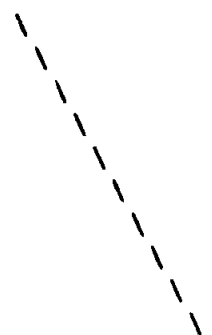
123456789012345678901234567890
123456789012345678901234567890

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.

816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869

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:

0000000000000000000000000000022
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
123456789012345678901234567890
123456789012345678901234567890

870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915

6.1.8 TEST 7 - OVERPRINT TEST

THIS TEST IS DESIGNED TO CHECK THE SPACING AND REPEATABLE PRINTING CHARACTERISTICS OF THE PRINTER. THREE ROWS OF CHARACTERS ARE EACH OVERPRINTED TWO TIMES. THE ROWS CONSIST OF THE FOLLOWING CHARACTERS ALTERNATED ACROSS THE LINE:

ROW 1 M-SP
ROW 2 SP-a
ROW 3 &-SP

THE RESULTING PATTERN WILL BE A CHECKERBOARD PATTERN AND THE OVERPRINTED CHARACTERS SHOULD BE ALIGNED PROPERLY WITH THE INITIAL CHARACTERS.

EXAMPLE:

M M M M M M M M M M M M M M M
a a a a a a a a a a a a a a a
& & & & & & & & & & & & & & &

WITH THE AUTO LINE FEED OPTION SET TO PRODUCE AN AUTOMATIC LINE FEED AFTER EVERY RECEIVED CARRIAGE RETURN, THE LINES WILL NOT BE OVERPRINTED. THERE WILL BE THREE LINES OF EACH CHARACTER WITH A BLANK LINE BETWEEN EACH GROUP OF CHARACTERS. THE CHARACTERS IN EACH GROUP SHOULD BE IN THE SAME COLUMNS.

EXAMPLE:

M M M M M M M M M M
M M M M M M M M M M
M M M M M M M M M M

a a a a a a a a a a
a a a a a a a a a a
a a a a a a a a a a

& & & & & & & & &
& & & & & & & & &
& & & & & & & & &

969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023

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:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

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:

```
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBB
```

1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077

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.

IF THE LA36 TERMINAL IS IN HALF DUPLEX WITH THE AUTO LINE FEED OPTION AVAILABLE, TYPING A CARRIAGE RETURN MAY CAUSE A GARBLED RESPONSE ON THE TERMINAL DURING THIS TEST.

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.

010

1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132

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

1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183

6.4 STANDARD I/O TESTS

THESE TESTS ARE DESIGNED AS A BRIEF CHECK OF THE CONSOLE TERMINAL INTERFACE LOGIC. EACH CHECK IS STRUCTURED AS AN INDEPENDENT TEST AND THE SWITCH REGISTER CONTROLS MAY BE USED. A DESCRIPTION OF EACH TEST IS GIVEN IN THE PROGRAM LISTING. ANY ERRORS ENCOUNTERED DURING THE I/O TESTS WILL CAUSE A HALT AT LOCATION 'ERRHLT' IF SWITCH 14 IS DOWN.

.TITLE CZLACFO LA36 TERM (DL11 & KL11)
:LA36 DIAGNOSTIC (DL11 & KL11 INTERFACE)
:AUTHORS: ROBERT W. BAKER
: R. QUENNEVILLE
: RALPH A. SCHAUER
: JOHN V. CHATALIAN
:
:COPYRIGHT 1974,1979 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

.SBTTL SWITCH REGISTER OPTIONS		
SWITCH	POSITION	FUNCTION
15	UP (1)	HALT AT COMPLETION OF CURRENT TEST
	DOWN (0)	CONTINUE NORMAL TEST SEQUENCE
14	UP (1)	CONTINUE ON ERROR
	DOWN (0)	HALT ON ERROR
13	UP (1)	DRIVE ONLY CONSOLE TERMINAL
	DOWN (0)	DRIVE ALL TERMINALS
11	UP (1)	LOOP ON INDIVIDUAL TEST
	DOWN (0)	NORMAL TEST SEQUENCE
09	UP (1)	CPU TYPE IS AN LSI-11, PDP-11/03
	DOWN (0)	ALL OTHER PDP-11 CPU'S
08	UP (1)	HALT TO SELECT TEST AT END OF CURRENT TEST
	DOWN (0)	LOOP ON TEST SEQUENCE
05-00		TEST # SELECTION
07-00		# OF COLUMNS AT START-UP

1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224

.SBTTL SPECIAL OPERATIONAL INFORMATION

- :1.-- THE STANDARD CONSOLE TERMINAL INTERRUPT VECTOR AND REGISTER ADDRESSES ARE USED. TO REDEFINE THE LOCATION OF THE CONSOLE TERMINAL THE SYMBOLIC LOCATIONS "CONADD" AND "CONVEC" SHOULD BE CHANGED BEFORE START UP.
- :2.-- BEFORE START UP REFER TO THE DESCRIPTION OF THE ROUTINE "DLY". TIMING IS A FUNCTION OF THE PDP-11 MODEL AND MEMORY TYPE AND SHOULD BE SET UP BEFORE RUNNING THE DIAGNOSTIC.
- :3.-- IF CPU IS A PDP-11/03 , LSI-11 SET SWITCH REGISTER BIT 09 TO A 1. SPECIAL TESTS ARE RUN ON THE DLV11.
- :4.-- SYSTEMS WITHOUT A HARDWARE SWITCH REGISTER SHOULD USE MEMORY LOCATION 176 AS A SOFTWARE SWITCH REGISTER.
- :5.-- THIS DIAGNOSTIC IS FOR VERIFICATION OF BASIC TERMINAL FUNCTIONS ONLY. IF THE TERMINAL UNDER TEST HAS HARDWARE OPTIONS INSTALLED RUN DIAGNOSTIC MAINDEC-11-DZLAF-A, THE LA36 TERMINAL OPTIONS TEST.
- :6.-- CONTROL G FUNCTION PERMITS CHANGING THE SOFTWARE SWITCH REGISTER AT ANY TIME AFTER A TEST. ALSO, AT TEST START, IF THERE IS NO HARDWARE SWITCH REGISTER, A MESSAGE WILL ASK FOR SOFTWARE SWITCH REGISTER SETTINGS WHICH MUST BE PROVIDED VIA TTY KEYBOARD. CONTROL U ALLOWS RESTARTING THE PROCESS IF AN ERROR WAS MADE. THE NUMBER MUST BE A SIX DIGIT OCTAL VALUE.

ECO HISTORY

- :CHGF1 - NEW STORAGE LOCATIONS FOR REFERENCE IN ^G MODE.
- :CHGF2 - CHANGE START UP FOR ENTERING SOFTWARE SWITCH REGISTER INFO.
- :CHGF3 - CHECK FOR ^G BETWEEN TESTS.
- :CHGF4 - CLEAR INTERRUPT ENABLE BIT TO PREVENT PREMATURE INTERRUPTS.
- :CHGF5 - ROUTINE TO CHECK PRINTOUT, UPDATE SOFTWARE SWITCH REG.
- :CHGF6 - NEW MESSAGES FOR SOFTWARE SWITCH REG ROUTINES.
- :CHGF7 - COMBINING LSI-11 TESTS TO ELIMINATE CLEARING DONE BIT IN TKS

```

1225          .SBTTL SYSTEM EQUATES
1226
1227          :
1228          :
1229          :
1230          000000          REGISTER EQUATES
1231          000001          R0=%0
1232          000002          R1=%1
1233          000003          R2=%2
1234          000004          R3=%3
1235          000005          R4=%4
1236          000006          R5=%5
1237          000007          SP=%6
1238          177776          PC=%7
1239          177776          PSW=177776
1240          :
1241          :
1242          :
1243          000001          SYSTEM EQUATES
1244          000002          BIT0=1
1245          000004          BIT1=2
1246          000010          BIT2=4
1247          000020          BIT3=10
1248          000040          BIT4=20
1249          000100          BIT5=40
1250          000200          BIT6=100
1251          000400          BIT7=200
1252          001000          BIT8=400
1253          002000          BIT9=1000
1254          004000          BIT10=2000
1255          010000          BIT11=4000
1256          020000          BIT12=10000
1257          040000          BIT13=20000
1258          100000          BIT14=40000
1259          000000          BIT15=100000
1260          040000          OPEN=0
1261          005726          SCOPSW=BIT14          ;SCOPE SWITCH
1262          022626          NITRSW=BIT11          ;TEST LOOP SWITCH
1263          000340          POPSP=5726           ;POP STACK ONCE
1264          000200          POPSP2=22626        ;POP STACK TWICE
1265          000200          PRTY7=340           ;PRIORITY LEVEL DEFINITIONS
1266          001000          PRTY4=200
1267          001000          ACRLF=200
1268          :
1269          :
1270          104000          PROGRAM TRAP EQUATES
1271          104001          TYPE=EMT+0
1272          104002          ERROR=EMT+1
1273          104003          EHALT=EMT+2
1274          104004          STRDRV=EMT+3
1275          104005          STPCHV=EMT+4
1276          104006          CHAIN=EMT+5
1277          104007          CHALT=EMT+6
1278          104010          TYPEM=EMT+7
1279          104011          DELAY=EMT+10
1280          104012          TTYCTL=EMT+11
          CRLF=EMT+12

```

1281	104013	SCRLF=EMT+13
1282	104014	Lr=EMT+14
1283	104015	PRINTC=EMT+15
1284	104016	PRTHDR=EMT+16
1285	104017	PRNT=EMT+17
1286	104020	READ=EMT+20
1287	104021	AREAD=EMT+21
1288	104022	CR=EMT+22
1289	104023	BTOASC=EMT+23
1290	104024	FORWD=EMT+24
1291	104025	READC EMT+25


```

1292
1293
1294
1295      000000
1296
1297
1298      000000
1299
1300 000000 000002
1301 000002 000000
1302 000004 000006
1303 000006 000000
1304 000010 000012
1305 000012 000000
1306 000014 000016
1307 000016 000000
1308 000020 000022
1309 000022 000000
1310 000024 000026
1311 000026 000000
1312 000030 003010
1313 000032 000340
1314
1315      000042
1316
1317 000042 000000
1318
1319      000046
1320
1321 000046 011524
1322
1323      000052
1324
1325 000052 010000
1326
1327      000174
1328
1329 000174 000000
1330 000176 000000
1331
1332 000200 000137 001026
1333 000204 000137 000754
1334 000210 000137 000772
1335 000214 000137 001010
1336
1337
1338      000600
1339
1340 000600 000000

```

.SBTTL TRAP CATCHER & STARTING ADDRESSES
 .ENABL ABS,AMA
 .ASECT
 MACHER: .+2 ;UNASSIGNED TRAP
 HALT ;SP OVERFLOW, BUS ERROR TRAP
 .+2 ;RESERVED INSTRUCTION TRAP
 HALT ;TRACE TRAP
 .+2 ;TRAP TO CALL IOX
 HALT ;POWER FAIL TRAP
 EMTINT ;EMT TRAP
 PRTY7
 .+42
 0
 .+46
 LOGICAL
 .+52
 010000
 .+174
 DISPREG: .WORD 0 ;SOFTWARE DISPLAY
 SWREG: .WORD 0 ;SOFTWARE SWITCH REGISTER
 JMP START ;START UP WITH I/O TESTS RUNNING
 JMP START1 ;START UP, SKIP ALL I/O TESTS
 JMP START2 ;START UP TERMINAL CONTROL WITH I/O TESTS
 JMP START3 ;START UP TERMINAL CONTROL WITHOUT I/O TESTS
 .+600
 SPBOT: 0 ;BOTTOM OF STACK

1341
1342
1343
1344
1345 000602 177560
1346 000604 000060
1347 000606 176500
1348
1349
1350
1351
1352 000610 000020
1353 000612 177560
1354 000614 177562
1355 000616 177564
1356 000620 177566
1357 000622 000060
1358 000624 000200
1359 000626 000064
1360 000630 000200
1361 000632 000000
1362 000634 000000
1363 000636 000000
1364 000640 000000
1365 000642 000000
1366 000644 000000
1367 000646 000000
1368 000650 000000
1369 000652 000000
1370 000654 000000
1371 000656 000000
1372 000660 000000
1373 000662 000000
1374 000664 000000
1375 000666 000000
1376 000670 000000
1377 000672 000251
1378 000674 000000
1379 000676 000000
1380 000700 000000
1381 000702 000000
1382 000704 000000
1383 000706 000000
1384 000710 000000
1385 000712 000000
1386 000714 177570
1387 000716 C00000
1388
1389 000720 177570
1390 000722 C00000
1391 000724 C00000
1392 000726 000000
1393 000730 000000
1394 000732 000000
1395 000734 000000
1396

.SBTTL SYMBOL DEFINITIONS
:
:
:
CONADD: 177560
CONVEC: 60
DLADR: 176500

DLNR: 16.
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
TKVTR: 60
TKLVL: PRTY4
TPVTR: 64
TPLVL: PRTY4
FSTD: OPEN
CNTLSW: OPEN
RTNNO: OPEN
NXTST: OPEN
SCOPIR: OPEN
PRGID: OPEN
CRBUF: OPEN
CTRA: OPEN
WIDTH: OPEN
LEVEL: OPEN
DLCNT: OPEN
ICTR: OPEN
REPT: OPEN
BRCTR: OPEN
COUNT3: OPEN
XCSR: OPEN
TIMER: 251
SPCNT: OPEN
CURTST: OPEN
TEMPCH: OPEN
PARITY: OPEN
PCHAR: OPEN
LFCNT: OPEN
INCHK: OPEN
TEMP: OPEN
SR: 177570
CNTR: OPEN
:***** CHGFI *****
DISPLAY: 177570
TIB: OPEN
TEMPST: OPEN
COUNT: OPEN
FILL: OPEN
MODE: OPEN
CNT: OPEN
:*****

: ADDR OF CONSOLE RECEIVER STATUS REG
: CONSOLE TERMINAL INTERRUPT VECTOR
: ADDRESS OF FIRST DL11, DEFAULT TO DL11-A,B
: IF DL11-C,D,E,, THEN
: SET TO 175610 FOR FIRST 16 (OF 31) OR
: SET TO 176000 FOR LAST 16 (OF 31)
: OR SET OTHER DESIRED START ADDRESS
: # OF DL11'S TO BE INITIALLY ASSUMED
: CONSOLE RECEIVER STATUS REG
: CONSOLE RECEIVER BUFFER
: CONSOLE TRANSMITTER STATUS REG
: CONSOLE TRANSMITTER BUFFER
: C.T. RECEIVER INTERRUPT VECTOR
: C.T. RECEIVER PRIORITY LEVEL
: C.T. TRANSMITTER INTERRUPT VECTOR
: C.T. TRANSMITTER PRIORITY LEVEL
: ADDRESS OF FIRST ACTIVE DL11
: CONSOLE TERMINAL CONTROL SWITCH
: CONTAINS CURRENT TEST NUMBER
: CONTAINS ADDRESS OF NEXT TEST
: CONTAINS ADDRESS OF TEST SCOPE ENTRY
: CONTAINS TEST PROGRAM INDICATORS

: CURRENT PAPER WIDTH, BINARY
: LEVEL OF EXECUTION
: # OF MULTIPLE DL11S
: I/O TEST ITERATION COUNT
: TEMP STORAGE FOR TESTS E021 & E022
: COUNTER FOR ROUTINE "AREAD"
: COUNTER FOR ROUTINE "PRINTC"
: ADDRESS OF MULTIPLE DL11 STATUS
: 1 MSEC COUNTER FOR ROUTINE "DELAY"
: COUNTER FOR TEST ROUTINE "PT3"
: ADDRESS OF CURRENT TEST
: TEMP STOR FOR ECHO TESTS
: PARITY FLAG FOR RECEIVED CHAR
: CHAR CODE WITH PARITY BIT
: COUNTER FOR TEST ROUTINE "PT4"
: CHECK FOR INPUT FLAG
: TEMPORARY WORKING STORAGE
: SW REG ADDRESS
: TIME COUNTER FOR LSI-11 TESTS

: DISPLAY REGISTER
: TEMPORARY KYBD BUFFER STORAGE
: TEMPORARY WORK LOCATION
: LOOP COUNT FOR TEST
: ZERO FILL SWITCH
: NUMBER OF DIGITS TO TYPE
: ITERATION COUNT

CZLACFO LA36 TERM (DL11 & KL11)
CZLACF.P11 13-JUN-79 15:35

MACY11 30A(1052) 13-JUN-79^{E 3} 15:35 PAGE 30
SYMBOL DEFINITIONS

SEQ 0050

1397

.SBTTL PROGRAM INITIALIZATION & CONTROL

```
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406 000736 005737 000654  
1407 000742 001403  
1408 000744 011600  
1409 000746 005740  
1410 000750 000000  
1411 000752 000002  
1412 000754 012737 177777 005356  
1413 000762 012737 104006 001772  
1414 000770 000424  
1415 000772 012737 104011 001772  
1416 001000 012737 005406 005356  
1417 001006 000415  
1418 001010 012737 104011 001772  
1419 001016 012737 177777 005356  
1420 001024 000406  
1421 001026 012737 005406 005356  
1422 001034 012737 104006 001772  
1423 001042 012706 000600  
1424  
1425 001046 013746 000006  
1426 001052 013746 000004  
1427 001056 012737 001076 000004  
1428  
1429 001064 022777 177777 177622  
1430 001072 001405  
1431 001074 000412  
1432 001076 022626  
1433 001100 012737 000202 000672  
1434 001106 012737 000176 000714  
1435 001114 012737 000174 000720  
1436 001122 012637 000004  
1437 001126 012637 000006  
1438 001132 022737 000176 000714  
1439 001140 001002  
1440 001142 004737 013752  
1441 001146 013701 000622  
1442 001152 012721 014176  
1443 001156 013721 000624  
1444 001162 005777 177426  
1445 001166 012777 000100 177416  
1446 001174 005037 000710  
1447 001200 012737 000006 000004  
1448 001206 005037 000644  
1449 001212 005037 000634  
1450 001216 005037 000654  
1451 001222 012737 003546 000024  
1452 001230 004737 003776  
1453
```

```
*****  
:COMMON HALT---WHEN IN SWITCH REGISTER CONTROL THE CPU  
: WILL BE ADVANCED TO THIS COMMON HALT WHERE  
: A NEW TEST WILL BE EXPECTED TO BE STARTED  
*****  
CHLT: TST LEVEL ;TEST CURRENT LEVEL  
BEQ SELHLT ;BRANCH IF 0, DO NOT HALT  
MOV @SP,R0 ;PUT ADDRESS OF CALLER INTO R0  
TST -(R0)  
HALT  
SELHLT: RTI ;RETURN FROM INTERRUPT  
START1: MOV #177777,ATOX ;FORCE END OF I/O TESTS  
MOV #CHALT,WAITF ;FORCE SR CONTROL  
BR STARTX  
START2: MOV #TTYCTL,WAITF ;FORCE TERMINAL CONTROL  
MOV #ATI,ATOX ;FORCE ALL I/O TESTS  
BR STARTX  
START3: MOV #TTYCTL,WAITF ;FORCE TERMINAL CONTROL  
MOV #177777,ATOX ;FORCE END OF I/O TESTS  
BR STARTX  
START: MOV #ATI,ATOX ;FORCE ALL I/O TESTS  
MOV #CHALT,WAITF ;FORCE SR CONTROL  
STARTX: MOV #SPBOT,SP ;SET STACK POINTER  
*****  
CHGF2: MOV 6,-(SP) ;SAVE CURRENT VECTOR  
MOV 4,-(SP)  
MOV #64$,@#4 ;SET UP LOC. 4 FOR NON-EXISTANT  
;MEMORY TRAP  
CMP #-1,@SR ;REFERENCE HARDWARE SWITCH REGISTER  
BEQ 65$ ;IF IT = -1,USE SOFT SW REG  
BR 66$ ;THEN USE HARDWARE SW REG  
64$: CMP (SP)+,(SP)+ ;CORRECT THE STACK, NO HDWE SW REG  
MOV #202,TIMER ;ADJUST TIMER FOR LSI-11  
65$: MOV #SWREG,SR ;POINT TO SOFT SW REG  
MOV #DISPREG,DISPLAY ;POINT TO SOFT DISP REG  
66$: MOV (SP)+,@#4 ;RESTORE VECTORS  
MOV (SP)+,@#6  
67$: CMP #SWREG,SR ;IS SOFT SWREG SELECTED  
BNE 60$ ;IF NOT,BR OVER NEXT OPER  
JSR PC,CNTLU ;SOFT SW REG INPUT ROUTINE  
60$: MOV TKVTR,R1  
MOV #TTINTS,(R1)+ ;VECTOR ADDRESS TO LOC 60  
MOV TKLVL,(R1)+ ;PRIORITY TO LOC 62  
TST @TKB ;CLEAR DONE FLAG  
MOV #100,@TKS ;SET TTY INTERRUPT ON  
CLR INCHK ;ALLOW INPUT CHECKING  
MOV #6,MACHER ;CLEAN UP  
CLR PRGID ;INITIALIZE PROGRAM FLAGS  
CLR CNTLSW ;INITIALIZE TERMINAL CONTROL SWITCH  
CLR LEVEL ;INITIALIZE LEVEL  
MOV #PFAIL,24 ;SET ADDR POWER FAIL ROUTINE  
JSR PC,CONIT ;SET UP CONSOLE TERMINAL ADDRESS  
*****
```

```

1454
1455
1456
1457
1458
1459
1460
1461
1462
1463 001234 017701 177454
1464 001240 042701 177400
1465 001244 020127 000204
1466 001250 003003
1467 001252 020127 000035
1468 001256 101002
1469 001260 012701 000204
1470 001264 010137 000652
1471 001270 012700 014627
1472 001274 012702 000003
1473 001300 104023
1474 001302 000401
1475 001304 000410
1476 001306 012700 000000
1477 001312 104015
1478 001314 104007
1479 001316 014376
1480 001320 012737 000240 001302
  
```

```

:*****
: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.
:THE SWITCHES MAY BE CHANGED ONCE THE PROGRAM TITLE OR THE DL11 COUNT
:MESSAGE HAS STARTED TO PRINT.
:*****
                                MOV    @SR,R1      ;PUT (SR) INTO R1
                                BIC    #177400,R1   ;SAVE ONLY BITS 0-7
                                CMP    R1,#204      ;TEST NO. COLUMN GT132
                                BGT    2$           ;COLUMNS GT132, DEFAULT TO 132
                                CMP    R1,#35       ;CHECK IF NO. COLUMNS LT 30
                                BHI    3$           ;NOT LT 30 NOR GT 132
                                MOV    #204,R1      ;COLUMNS LT 30 OR GT 132, DEFAULT
                                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
                                BR     5$
                                BR     6$
                                MOV    #0,P0       ;TRANSMIT A
                                PRINTC              ;NULL CODE
                                TYPEM              ;TYPE PROGRAM TITLE FIRST TIME RUN
                                STARTM
                                MOV    #NOP,4$
  
```

```
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488 001326 012737 001406 000004 6$: MOV #END2A,MACHER ;INITIALIZE TIME OUT TRAP  
1489 001334 013700 000606 MOV DLADR,R0 ;ADDRESS OF FIRST DL11 TO R0  
1490 001340 013701 000610 MOV DLNR,R1 ;SET DL CHECK COUNT  
1491 001344 005037 000656 CLR DLCNT ;INITIALIZE DLCNT  
1492 001350 005710  
1493 001352 012737 001420 000004 END3: TST (R0) ;IS DL PRESENT?  
1494 001360 010037 000632 MOV #END2,MACHER ;YES, RESET TIME OUT TRAP  
1495 001364 000401 MOV RO,FSTDL ;STORE ADDRESS OF FIRST DL11  
1496 001366 005710 BR 2$ ;CONTINUE  
1497 001370 062700 000010 1$: TST (R0) ;IS DL11 PRESENT  
1498 001374 005237 000656 2$: ADD #10,R0 ;POINTER AND DL11 ADDRESS  
1499 001400 005301 INC DLCNT ;INCREMENT COUNT OF DL11'S  
1500 001402 001407 DEC R1 ;DECREMENT DL CHECK COUNT, DONE?  
1501 001404 000770 BEQ END4 ;BRANCH IF DONE  
1502 001406 005301 BR 1$ ;CHECK PRESENCE OF NEXT DL11  
1503 001410 001404 END2A: DEC R1 ;DONE DL CHECK?  
1504 001412 062700 000010 BEQ END4 ;YES, EXIT  
1505 001416 000754 ADD #10,R0 ;NO, CHECK NEXT DL  
1506 001420 022626 BR END3 ;CONTINUE  
1507 001422 013701 000656 END2: POPSP2 ;DL11 NOT PRESENT  
1508 001426 012700 014564 END4: MOV DLCNT,R1 ;GET # DL11'S  
1509 001432 012702 000002 MOV #DL11S1,R0 ;ADR OF ASCII CHAR STORAGE  
1510 001436 104023 MOV #2,R2 ;# OF ASCII CHARS  
1511 001440 104007 BTOASC ;CONVERT NUMBER  
1512 001442 014551 TYPEN ;TYPE MESSAGE  
1513 DL11S  
1514  
1515  
1516  
1517  
1518  
1519  
1520 001444 005037 000636 CLR RTNNO ;SET ROUTINE NO - 0  
1521 001450 005037 000654 CLR LEVEL ;SET LEVEL - 0  
1522 001454 023727 005356 177777 CMP ATOX,#177777 ;SEE IF I/O IS TO BE SKIPPED  
1523 001462 001517 BEQ SKIP  
1524 001464 012737 005354 000640 MOV #ATO,NXTST ;ADDRESS OF FIRST I/O TEST  
1525 001472 104024 FORWD ;SET UP TEST PARAMETERS  
1526 001474 000177 177176 JMP @CURTST ;GO TO I/O TEST ROUTINE
```

```

1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549 001500
1550 001500 004737 013700
1551 001504 032737 000001 000634
1552 001512 001401
1553 001514 104011
1554 001516 005737 000644
1555 001522 100016
1556 001524 032777 040000 177162
1557 001532 001407
1558 001534 022737 177777 000642
1559 001542 001403
1560 001544 017716 177072
1561 001550 000002
1562 001552 042737 100000 000644
1563 001560 005737 000654
1564 001564 001405
1565 001566 032777 004000 177120
1566 001574 001405
1567 001576 000002
1568 001600 005337 000660
1569 001604 001407
1570 001606 000002
1571 001610 032777 000400 177076
1572 001616 001402
1573 001620 000137 001772
1574 001624 022626
1575 001626 000240
1576 001630 005777 177060
1577 001634 100003
1578 001636 113700 000636
1579 001642 000000
1580 001644 005737 000654
1581 001650 001420
1582 001652 012737 000006 000004

```

```

*****
:CHAINN-- THIS PORTION IS THE COMMON RETURN
:          FOR ALL THREE CLASSES OF TESTS.
:
:          1--IF AN ERROR OCCURRED DURING AN I/O TEST THE
:          OPERATOR CAN CAUSE THAT TEST TO BE LOOPED
:          WITHOUT ANY FURTHER ERROR HALTS BY
:          SETTING THE "SCOPE" BIT (#14) ON THE SR=1.
:          RESETTNG SR BIT 14 TO 0 WILL ALLOW THE
:          ERROR HALT TO OCCUR AGAIN IF IT STILL EXISTS.
:
:          2--IF THE OPERATOR IS IN THE MAINTENANCE MODE, (BIT 8=1
:          AT START UP TIME), AND BIT 11=1, THE SELECTED PROGRAM WILL
:          FIRST HALT. THEN HIT CONTINUE AND PROGRAM WILL LOOP CONTINUOUSLY.
:          IF BIT 8=0, AND BIT 11=0, THE SELECTED PROGRAM WILL BE
:          ADVANCED TO THE NEXT TEST IN ITS CLASS. AS LONG AS BITS
:          8 AND 11 = 0, THE CLASS OF TESTS SELECTED WILL BE CON-
:          UOUSLY SEQUENCED THROUGH.
:          IF BIT 8=1, AND BIT 11=0, THEN THE CPU WILL HALT AT LOC
:          SELHLT AND WAIT FOR THE NEXT TEST NUMBER TO BE SET IN
:          THE SWITCH REGISTER.

```

```

*****
CHGF3:
CHAINN: JSR      PC,CKSWR      ;SEE IF A ^G HAS BEEN GIVEN
        BIT      #1,CNTLSW    ;CHECK IF TERMINAL CONTROL
        BEQ      1$          ;BRANCH IF NOT
        TTYCTL   ;GO TO TERMINAL CONTROL
1$:     TST      PRGID        ;TEST ERROR BIT IN PRGID
        BPL      3$          ;BRANCH IF ERROR BIT NOT SET
        BIT      #SCOPSW,@SR  ;ERROR, CHECK IF SCOPE OPTION ON
        BEQ      2$          ;BRANCH IF NO SCOPING
        CMP      #-1,SCOPTR   ;YES, CHECK IF OK TO SCOPE THIS TEST
        BEQ      2$          ;BRANCH IF NOT OK
        MOV      @SCOPTR,@SP  ;PUT ADDR OF SCOPE ENTRY INTO STACK
        RTI      ;GO TO SCOPE ENTRY IN TEST
2$:     BIC      #BIT15,PRGID  ;CLEAR ERROR IND. IN PRGID
3$:     TST      LEVEL        ;CHECK LEVEL
        BEQ      4$          ;BRANCH IF LEVEL=0
        BIT      #NITRSW,@SR  ;TEST LOOP SWITCH ON (-1)
        BEQ      5$          ;BRANCH IF NO LOOP TEST
        RTI      ;GO BACK TO TEST
4$:     DEC      ICTR         ;DECREMENT TEST ITERATION COUNT
        BEQ      6$          ;BRANCH IF COUNT=0
        RTI      ;NOT ZERO, REPEAT TEST
5$:     BIT      #BIT8,@SR    ;TEST IF SEQUENCE TEST (BIT8)
        BEQ      6$          ;BRANCH TO NEXT TEST IF BIT8 0
        JMP      WAITF       ;GO WAIT FOR MORE INPUT
6$:     POPSP2   ;POP 2 OFF STACK
CHAINY: NOP                ;THIS FORMERLY WAS RESET
        TST      @SR         ;CHECK SR
        BPL      1$          ;BRANCH IF NO HALT WANTED
        MOVB    RTNNO,RO     ;CURRENT TEST NUMBER TO RO
        HALT    ;HALT (NOT FOR TEST SELECTION)
1$:     TST      LEVEL        ;TEST THE CURRENT LEVEL
        BEQ      3$          ;BRANCH IF 0
        MOV      #6,MACHER   ;CLEAN UP

```



```

1583 001660 012706 000600      MOV      #SPBOT,SP      ;SET UP STACK POINTER
1584 001664 104024      FORWD     ;SET UP VALUES FOR NEXT TEST
1585 001666 022737 177777 000640  CMP      #-1,NXTST      ;END OF I/O TESTS (=-1)
1586 001674 001004      BNE      2$             ;BRANCH IF NOT END
1587 001676 012737 005354 000640  MOV      #ATO,NXTST      ;RESET NXTST TO FIRST I/O TEST
1588 001704 104024      FORWD     ;SET UP VALUES FOR NEXT TEST
1589 001706 000177 176764      2$:      JMP      @CURTST        ;GO TO TEST
1590 001712 022737 177777 000640  3$:      CMP      #-1,NXTST      ;END OF I/O TESTS (=-1)
1591 001720 001012      BNE      NEXT          ;BRANCH IF NOT
1592 001722 032777 000400 176764  SKIP:    BIT      #BIT8,@SR    ;TEST IF WANT TEST SELECTION RIGHT AWAY
1593 001730 001016      BNE      NEXT1         ;BRANCH IF NOT
1594 001732 052737 000200 000644  BIS      #BIT7,PRGID     ;BYPASS SCOPING
1595 001740 012737 007374 000640  MOV      #PTO,NXTST      ;PROD TESTING, GO TO PRINTER TESTS
1596 001746 012737 000006 000004  NEXT:    MOV      #6,MACHER ;CLEAN UP
1597 001754 012706 000600      MOV      #SPBOT,SP      ;SET UP STACK POINTER
1598 001760 104024      FORWD     ;SET UP NEXT TEST PARAMETERS
1599 001762 000177 176710      JMP      @CURTST        ;GO TO ROUTINE
1600 001766 005237 000654  NEXT1:   INC      LEVEL

1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612 001772 104006      WAITF:   CHALT          ;OR TTYCTL IF START WAS AT 210
1613 001774 012737 000006 000004  MOV      #6,MACHER      ;CLEAN UP
1614 002002 012706 000600      MOV      #SPBOT,SP      ;SET UP STACK POINTER
1615 002006 017700 176702      MOV      @SR,R0         ;GET CURRENT SW REG
1616 002012 042700 177700      BIC      #177700,R0
1617 002016 02C027 000037      CMP      R0,#37         ;TEST IF PROG NO. IS I/O TEST
1618 002022 101403      BLOS    1$             ;BRANCH IF EQ OR LT 37. AN ECHO OR PRINTER
1619 002024 005037 000644      CLR      PRGID         ;I/O TEST, CLEAR PRGID
1620 002030 000403      BR      2$
1621 002032 052737 000200 000644  1$:      BIS      #BIT7,PRGID     ;BYPASS SCOPING
1622 002040 000241      2$:      CLC                    ;CLEAR C BIT
1623 002042 006100      ROL      R0             ;GET PROGRAM ADDRESS OUT OF
1624 002044 016037 002614 000640  MOV      PRGTAB(R0),NXTST;PROGRAM ADDRESS TABLE
1625 002052 023727 000640 001772  CMP      NXTST,#WAITF    ;TEST IF LEGAL TEST NO.
1626 002060 001744      BEQ     WAITF          ;BRANCH IF ILLEGAL
1627 002062 104024      FORWD     ;SET UP TEST PARAMETERS
1628 002064 000177 176606      JMP      @CURTST        ;GO TO TEST

```

1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1640
 1641
 1642 002070 022626
 1643 002072 105777 176514
 1644 002076 100013
 1645 002100 017705 176510
 1646 002104 042705 177600
 1647 002110 020527 000177
 1648 002114 001004
 1649 002116 042737 0044C0 000634
 1650 002124 000413
 1651 002126 032737 004000 000634 1\$:
 1652 002134 001401
 1653 002136 000002
 1654 002140 032737 000400 000634 2\$:
 1655 002146 001402
 1656 002150 000137 001626
 1657 002154 012737 177777 000710 TTY1B:
 1658 002162 012700 000036
 1659 002166 104010
 1660 002170 104007
 1661 002172 015041
 1662 002174 005037 000710
 1663 002200 104020 1\$:
 1664 002202 023727 000700 000040
 1665 002210 001773
 1666 002212 012700 000036
 1667 002216 104010
 1668 002220 104017
 1669 002222 117777 176366 176370
 1670 002230 004737 002552
 1671 002234 000541
 1672 002236 010005
 1673 002240 006305
 1674 002242 006305
 1675 002244 006305
 1676 002246 104020 2\$:
 1677 002250 023727 000700 000040
 1678 002256 001773
 1679 002260 012700 000036
 1680 002264 104010
 1681 002266 104017
 1682 002270 117777 176320 176322
 1683 002276 004737 002552
 1684 002302 000516

 TTY1-- THIS SECTION IS USED WHEN THE DIAGNOSTIC IS BEING CONTROLLED BY
 THE CONSOLE TERMINAL. IT IS EFFECTIVE ONLY WHEN THE DIAGNOSTIC
 STARTING ADDRESS IS 210 AND SR BIT 8 WAS SET AT START TIME.
 THE RESPONSE TO THE MESSAGE "SELECT TEST NO." MUST BE THE 2
 DIGIT OCTAL TEST NUMBER FOLLOWED BY :
 "L" TO LOOP ON TEST
 "S" TO LOOP ON SEQUENCE
 "." TO EXECUTE TEST ONCE
 ALL SPACES WILL BE IGNORED. AN ILLEGAL INPUT WILL BE FLAGGED BY A "?"
 AND THE RETYPING OF THE ABOVE MESSAGE.

TTY1: POPSP2 ;POP 2 FROM STACK
 TSTB @TKS ;TEST IF ANY INPUT
 BPL 1\$;BRANCH IF NOT
 MOV @TKB,R5 ;GET CHAR
 BIC #177600,R5 ;MASK BITS
 CMP R5,#177 ;CHECK IF RUBOUT
 BNE 1\$;BRANCH IF NOT
 BIC #4400,CNTLSW ;CLEAR LOOP BITS
 BR TTY1B
 BIT #NITRSW,CNTLSW ;CHECK IF LOOP ON TEST
 BEQ 2\$;BRANCH IF NO LOOP ON TEST
 RTI ;LOOP ON TEST
 BIT #BIT8,CNTLSW ;TEST IF LOOP ON SEQUENCE
 BEQ TTY1B ;BRANCH IF NO LOOP ON SEQUENCE
 JMP CHAINY ;CHAIN TO NEXT TEST
 TTY1B: MOV #-1,INCHK ;STOP INPUT CHECKING
 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
 DELAY
 TYPEM
 MMSG3 ;TYPE MESSAGE
 CLR INCHK ;ALLOW INPUT CHECKING AGAIN
 1\$: READ ;WAIT FOR INPUT
 CMP TEMPCH,#40 ;TEST IF CHAR IS A SPACE
 BEQ 1\$;BRANCH IF YES
 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
 DELAY
 PRNT ;READY?
 MOV @TKB,@TPB ;ECHO CHAR
 JSR PC,TESTC ;CHECK IF CHAR IS OK
 BR 8\$;NO, ERROR
 MOV R0,R5 ;OK, PUT CHAR INTO R5
 ASL R5 ;SHIFT INTO POSITION 5-3
 ASL R5
 ASL R5
 2\$: READ ;WAIT FOR NEXT CHAR
 CMP TEMPCH,#40 ;CHECK IF A SPACE
 BEQ 2\$;BRANCH IF SPACE
 MOV #30.,R0 ;DELAY FOR HALF DUPLEX
 DELAY
 PRNT ;READY?
 MOV @TKB,@TPB ;ECHO CHAR
 JSR PC,TESTC ;CHECK IF CHAR IS OK
 BR 8\$;ERROR IN CHAR


```

1729
1730
1731 002552 023727 000700 000060
1732 002560 103001
1733 002562 000207
1734 002564 023727 000700 000067
1735 002572 101401
1736 002574 000207
1737 002576 062716 000002
1738 002602 013700 000700
1739 002606 042700 177770
1740 002612 000207

```

:TESTC--CHECKS THAT THE INPUTTED CHARACTER IS BETWEEN 0 AND 7 INCLUSIVE

```

TESTC:  CMP      TEMPCH,#60      :CHECK IF NUMERIC AND EQ OR GT 0
        BHIS     1$              :BRANCH IF OK
        RTS      PC              :ERROR RETURN
1$:     CMP      TEMPCH,#67      :CHECK IF EQ OR LT 7
        BLOS     2$              :BRANCH IF OK
        RTS      PC              :ERROR RETURN
2$:     ADD      #2,@SP           :SET UP RETURN ADDRESS
        MOV      TEMPCH,R0       :GET CHAR
        BIC      #177770,R0      :SAVE ONLY THE DIGIT
        RTS      PC              :NORMAL RETURN

```


1797	002774	001772	
1798	002776	001772	
1799	003000	001772	
1800	003002	001772	
1801	003004	001772	
1802	003006	001772	
1803			
1804			
1805			
1806			
1807			
1808			
1809	003010	011646	
1810	003012	162716	000002
1811	003016	017616	000000
1812	003022	121627	000035
1813	003026	101402	
1814	003030	000000	
1815	003032	000776	
1816	003034	006116	
1817	003036	042716	177001
1818	003042	062716	003064
1819	003046	017616	000000
1820	003052	005046	
1821	003054	012746	003062
1822	003060	000002	
1823	003062	000136	
1824			
1825	003064	003164	
1826	003066	003406	
1827	003070	003434	
1828	003072	003444	
1829	003074	003474	
1830	003076	001500	
1831	003100	000736	
1832	003102	003252	
1833	003104	003524	
1834	003106	002070	
1835	003110	003302	
1836	003112	003230	
1837	003114	003304	
1838	003116	004412	
1839	003120	003324	
1840	003122	004402	
1841	003124	004200	
1842	003126	003726	
1843	003130	003314	
1844	003132	004074	
1845	003134	003650	
1846	003136	004272	
1847	003140	003160	
1848	003142	003160	
1849	003144	003160	
1850	003146	003160	
1851	003150	003160	
1852	003152	003160	

```
WAITF : SPARE
WAITF : SPARE
WAITF : SPARE
WAITF : SPARE
WAITF : SPARE
WAITF : SPARE
WAITF : SPARE

:*****
:EMTINT -----SERVICE ROUTINE FOR TRAPS THROUGH
:                               LOCATION 30.
:*****

EMTINT: MOV @SP,-(SP) ;PUSH STACKED PC TO GET A WORK COPY. (Q)
SUB #2,@SP ;SUB 2 TO POINT TO CALLING TRAP INSTR.
MOV @SP,@SP ;PLACE TRAP INSTR INTO THIS STACK WORK AREA.
CMPB @SP,#35 ;EXAMINE ITS RIGHT SIDE. (Q)
BLOS 2$ ;BRANCH IF WITHIN RANGE OF ESTABLISHED TABLE.
1$: HALT ;ELSE HALT.
BR 1$
2$: ROL @SP ;MULT INSTR BY 2 TO GET WORD DISPLACEMENT.
BIC #177001,@SP ;STRIP OFF OP CODE AND LS BIT.
ADD #EMTTAB,@SP ;ADD IN STARTING ADDRESS OF TABLE.
MOV @SP,@SP ;FROM TABLE GET OUT DESIRED POINTER.
CLR -(SP) ;PUSH A ZERO PSW.
MOV #3$,-(SP) ;PUSH A PC = TO #3$ OF THIS ROUTINE.
RTI ;DO RTI (POP-POP) TO ESTABLISH THE ZERO PSW.
3$: JMP @SP+ ;JMP TO ROUTINE LEAVING STACK AS FOUND.

EMTTAB: TYP ;MESSAGE OUTPUT ROUTINE
ERR ;I/O TEST ERROR ROUTINE
EHLT ;UNCONDITIONAL HALT
STLSRV ;KEYBOARD VECTOR/PRIORITY SETUP
STLSPV ;PRINTER VECTOR/PRIORITY SETUP
CHAINN ;COMMON TEST EXIT
CHLT ;SR BIT 15 HALT
TYPM ;MESSAGE OUTPUT ROUTINE, MULTI DEVICES
DLY ;DELAY ROUTINE
TTY1 ;CONSOLE TERMINAL CONTROL
$CRLF ;CARRIAGE RETURN-LINE FEED TO ALL DL11'S
$$CRLF ;CARRIAGE RETURN-LINE FEED TO CONSOLE
$LF ;LINE FEED ONLY (TO ALL)
$PRTC ;PRINT CHAR
$PRHDR ;PRINT TEST HEADER
$PRNT ;PRINTER READY
$READ ;READ CHAR
$AREAD ;I/O TEST READ ROUTINE
$CR ;CARRIAGE RETURN ONLY (TO ALL)
$BTASC ;BINARY TO ASCII CONVERSION
$FORWD ;FORWARD ROUTINE ( BETWEEN TESTS )
$READC ;READ CONSOLE KYBD ONLY
SPARET ;SPARE EMT
SPARET ;SPARE EMT
SPARET ;SPARE EMT
SPARET ;SPARE EMT
SPARET ;SPARE EMT
SPARET ;SPARE EMT
SPARET ;SPARE EMT
```

1853 003154 003160
 1854 003156 003160
 1855 003160 000000
 1856 003162 000776
 1857
 1858
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1870
 1871
 1872
 1873
 1874
 1875
 1876
 1877
 1878
 1879 003164 010046
 1880 003166 016601 000002
 1881 003172 062766 000002 000002
 1882 003200 011101
 1883 003202 112100
 1884 003204 100403
 1885 003206 001004
 1886 003210 012600
 1887 003212 000002
 1888 003214 104013
 1889 003216 000771
 1890 003220 104017
 1891 003222 110077 175372
 1892 003226 000765
 1893
 1894 003230 104017
 1895 003232 112777 000015 175360
 1896 003240 104017
 1897 003242 112777 000012 175350
 1898 003250 000002

SPARET SPARET : SPARE EMT
 SPARET SPARET : SPARE EMT
 SPARET: HALT : HALT IF TRAP TO UNDEFINED
 BR SPARET : EMT IS ATTEMPTED.

.SBTTL COMMON ROUTINES USED BY LA36 TESTS

```

:*****
:THIS SECTION CONTAINS MOST ROUTINES CALLED BY
:THE VARIOUS TESTS EITHER BY TRAPPING THROUGH LOCATION
:30 OR BY SUBROUTINE CALLS (JSR PC,***).
:*****
:TYPE-- A COMMON ROUTINE USED TO TYPE MESSAGES ON THE
:CONSOLE TERMINAL ONLY. THE NULL CHARACTER TERMINATES
:THE MESSAGE. CALLED THROUGH AN EMT TRAP.
:CALLING SEQUENCE
:TYPE
:MSG      :ADDRESS OF MESSAGE
:*****
TYP:  MOV  RO,-(SP)      :SAVE RO
      MOV  2(SP),R1     :GET POINTER TO ADDR. OF MSG.
      ADD  #2,2(SP)
      MOV  (R1),R1      :ADDR. OF MSG TO R1
1$:   MOVB (R1)+,RO     :GET CHAR
      BMI  2$          :BRANCH IF WANT AUTO CR-LF
      BNE  3$          :PRINT CHAR IF NOT NULL
      MOV  (SP)+,RO     :RESTORE RO
      RTI              :EXIT IF NULL CHAR
2$:   SCRLF           :YES, SEND CR-LF
      BR   1$          :GET NEXT CHAR
3$:   PRNT            :PRINTER READY?
      MOVB RO,@TPB     :LOAD PRINTER BUFFER WITH CHAR
      BR   1$          :GO GET NEXT CHAR
$SCRLF: PRNT          :PRINTER READY?
      MOVB #15,@TPB    :SEND CR
      PRNT            :PRINTER READY?
      MOVB #12,@TPB    :SEND LF
      RTI              :RETURN TO CALLER
  
```

1899			
1900			
1901			
1902			
1903			
1904			
1905			
1906			
1907			
1908			
1909	003252	011601	
1910	003254	062716	000002
1911	003260	011101	
1912	003262	112100	
1913	003264	100402	
1914	003266	001003	
1915	003270	000002	
1916	003272	104012	
1917	003274	000772	
1918	003276	104015	
1919	003300	000770	
1920			
1921	003302	104022	
1922	003304	012700	000012
1923	003310	104015	
1924	003312	000002	
1925			
1926	003314	012700	000015
1927	003320	104015	
1928	003322	000002	
1929			
1930			
1931			
1932			
1933			
1934			
1935			
1936	003324	012700	000000
1937	003330	104015	
1938	003332	104007	
1939	003334	014613	
1940	003336	013700	000636
1941	003342	006200	
1942	003344	006200	
1943	003346	006200	
1944	003350	042700	177770
1945	003354	062700	000060
1946	003360	104015	
1947	003362	013700	000636
1948	003366	042700	177770
1949	003372	062700	000060
1950	003376	104015	
1951	003400	104012	
1952	003402	104014	
1953	003404	000002	

```

:XXXXXXXXXX
:TYPM---MULTI TYPE-A COMMON ROUTINE TO OUTPUT
: A MESSAGE ON ALL DL11S IF THE MULTI TEST
: SWITCH (BIT 13) IS RESET. THIS ROUTINE IS USED BY
: THE PRINTER TESTS TO TYPE HEADINGS. IF A UNIT
: IS NOT READY, THE CHARACTER WILL NOT BE TYPED.
:XXXXXXXXXX
TYPM:  MOV      (SP),R1      ;GET POINTER TO ADDR OF MMSG
      ADD      #2,@SP
      MOV      (R1),R1      ;ADDR OF MMSG TO R1
1$:    MOVB    (R1)+,R0      ;GET CHAR
      BMI     2$            ;BRANCH IF WANT AUTO CR-LF
      BNE     3$            ;CONTINUE IF NOT NULL
      RTI
2$:    CRLF
      BR      1$           ;YES, SEND CR-LF
3$:    PRINTC 1$           ;NEXT CHAR
      BR      1$           ;PRINT CHAR
      BR      1$           ;GO GET NEXT CHAR.

$CRLF: CR          ;SEND CR
$LF:   MOV      #12,R0    ;SET LF CHAR
      PRINTC
      RTI              ;RETURN TO CALLER

$CR:   MOV      #15,R0    ;SET CR CHAR
      PRINTC
      RTI              ;SEND IT
      RTI              ;RETURN

:*****
:ROUTINE TO PRINT TEST HEADER
:*****

$PRHDR: MOV      #0,R0    ;TRANSMIT
      PRINTC          ;NUL CODE.
      TYPM           ;PRINT MESSAGE
      HDRMSG
      MOV      RTNNO,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 DIGIT
      MOV      RTNNO,R0   ;GET TEST NUMBER AGAIN
      BIC     #177770,R0  ;MASK LAST DIGIT
      ADD     #60,R0     ;MAKE ASCII
      PRINTC          ;PRINT DIGIT
      CRLF          ;CR-LF
      LF           ;BLANK LINE
      RTI           ;RETURN
  
```



```

1954
1955
1956
1957
1958
1959
1960
1961
1962
1963 003406 032777 040000 175300 ERR: BIT #SCOPSW,@SR ;CHECK SCOPE SWITCH
1964 003414 001404 BEQ 1$ ;BRANCH IF NO SCOPE
1965 003416 005737 000644 TST PRGID ;SCOPING WANTED, FIRST ERROR?
1966 003422 100001 BPL 1$ ;BRANCH AND HALT ON FIRST ERROR
1967 003424 000002 RTI ;SCOPE EXIT
1968 003426 052737 100000 000644 1$: BIS #BIT15,PRGID ;SET ERROR INDICATOR
1969 003434 011600 EHLT: MOV @SP,R0
1970 003436 005740 TST -(R0) ;ADDRESS OF CALL INTO R0
1971 003440 000000 HALT
1972 003442 000002 ERRHLT: RTI ;RETURN TO TEST FOLLOWING CALL
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984 003444 017637 000000 003464 STLSRV: MOV @(SP),STPRA+2 ;SET RETURN ADR AND VECTOR
1985 003452 062716 000002 ADD #2,@SP
1986 003456 013701 000622 MOV TKVTR,R1
1987 003462 012721 000000 STPRA: MOV #0,(R1)+
1988 003466 013721 000624 MOV TKLVL,(R1)+
1989 003472 000002 RTI
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 003474 017637 000000 003514 STLSPV: MOV @(SP),STPPA+2 ;SET RETURN ADR AND VECTOR
2002 003502 062716 000002 ADD #2,@SP
2003 003506 013701 000626 MOV TPVTR,R1
2004 003512 012721 000000 STPPA: MOV #0,(R1)+
2005 003516 013721 000630 MOV TPLVL,(R1)+
2006 003522 000002 RTI ;RETURN TO CALLER
  
```

```

:*****
:ERRA-- COMMON ERROR RETURN FROM I/O TESTS. HALTS
: WITH ADDRESS OF ERROR IN R0. TO CONTINUE
: ON SAME TEST BUT NOT HALTING ON ERROR,
: SET THE SCOPE BIT (14) - 1 AND PRESS CONTINUE
:*****
  
```

```

:*****
:STLSRV--- THIS ROUTINE SETS UP KEYBOARD INTERRUPT
: VECTOR AND PRIORITY. CALLING SEQUENCE
  
```

```

: STRDRV
: AT20C ;LOCATION OF NEW INTERRUPT VECTOR
:*****
  
```

```

:*****
:STLSPV-- THIS ROUTINE SETS UP PRINTER INTERRUPT
: VECTOR AND PRIORITY CALLING SEQUENCE
: STPCHV
: AT35E ;LOCATION OF NEW INTERRUPT VECTOR
:*****
  
```

2007
 2008
 2009
 2010
 2011
 2012
 2013
 2014
 2015
 2016
 2017
 2018
 2019
 2020
 2021
 2022
 2023
 2024
 2025
 2026
 2027
 2028
 2029
 2030
 2031
 2032
 2033
 2034
 2035
 2036
 2037
 2038
 2039
 2040
 2041
 2042

003524 010146
 003526 013701 000672
 003532 005301
 003534 001376
 003536 005300
 003540 001372
 003542 012601
 003544 000002

```

*****
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 11/05,11/10(=251).
  
```

MACHINE	05&10	35&40	15&20	LS1&03	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=	5.9USEC	2.75	4.9	7.7	.90USEC		1.49USEC	
SET TIMER	251	554	314	202	2127		1237	755

2.03USEC

```

XXXXXXXXXX
DLY:  MOV R1,-(SP) ;SAVE R1
1$:   MOV TIMER,R1 ;MOV 1 MSEC LOOP CNT TO R1
2$:   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
  
```

2043
2044
2045
2046
2047
2048
2049

:*****
:PFAIL--POWER FAIL ROUTINE
: SAVE ALL REGISTERS AND SET RESTART ADDRESS
: INTO LOCATION 24
:RESTART--POWER FAIL RECOVERY

CZLACFO LA36 TERM (DL11 & KL11)
CZLACF.P11 13-JUN-79 15-33

MACY11 30A(1052) 13-JUN-79^{H 4} 15:35 PAGE 46
COMMON ROUTINES USED BY LA36 TESTS

SEQ 0046

2050

: RESTORE ALL REGISTERS AND GO TO START

!

```
2051
2052
2053
2054 003546 010046
2055 003550 010146
2056 003552 010246
2057 003554 010346
2058 003556 010446
2059 003560 010546
2060 003562 013746 000024
2061 003566 010637 003602
2062 003572 012737 003604 000024
2063 003600 000000
2064 003602 000000
2065 003604 104007
2066 003606 003640
2067 003610 013706 003602
2068 003614 012637 000024
2069 003620 012605
2070 003622 012604
2071 003624 012603
2072 003626 012602
2073 003630 012601
2074 003632 012600
2075 003634 000137 001026
2076
2077 003640 050200 053517 051105 1$:
2078 003646 000200
2079
```

```

:
:*****
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:  .TYPEM
        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    START
        .ASCIZ <ACRLF>/POWER/<ACRLF>
        .EVEN
```

2080
 2081
 2082
 2083
 2084
 2085
 2086
 2087
 2088
 2089
 2090
 2091
 2092 003650 013705 000640
 2093 003654 012537 000636
 2094 003660 012537 000640
 2095 003664 105737 000644
 2096 003670 100407
 2097 003672 012537 000660
 2098 003676 012537 000642
 2099 003702 010537 000676
 2100 003706 000002
 2101 003710 012737 177777 000642
 2102 003716 012737 000001 000660
 2103 003724 000766

```

:*****
:FORWARD--THIS ROUTINE TRANSFERS THE 2 OR 4 ARGUMENTS
:FROM THE TEST ROUTINE. THEY ARE:
:
:1- ROUTINE NUMBER
:2- ADDRESS OF NEXT TEST
:3- ITERATION COUNT (I/O TESTS ONLY)
:4- SCOPE ENTRY ADDRESS (I/O TESTS ONLY)
:*****
  
```

```

$FORWD: MOV     NXTST,R5           ;ADDR OF NEXT TEST TO R5
          MOV     (R5)+,RTNNO      ;GET NUMBER OF NEXT TEST
          MOV     (R5)+,NXTST      ;GET ADDR OF FOLLOWING TEST
          TSTB   PRGID             ;CHECK IF I/O TEST
          BMI    FORWDB           ;SKIP THE FETCH OF ITER CNT AND SCOPE
          MOV     (R5)+,ICTR       ;GET ITERATION COUNT
          MOV     (R5)+,SCOPTR     ;GET SCOPE ENTRY POINT
FORWDA:  MOV     R5,CURTST         ;ENTRY POINT TO TEST IN CUR TST
          RTI
FORWDB:  MOV     #-1,SCOPTR        ;FORCE NO SCOPE
          MOV     #1,ICTR          ;FORCE INTERATION COUNT OF 1
          BR     FORWDA
  
```

2104
 2105
 2106
 2107
 2108
 2109
 2110
 2111
 2112
 2113
 2114
 2115
 2116 003726 012737 000600 000664
 2117 003734 052777 000004 174654
 2118 003742 005077 174652
 2119 003746 105777 174640
 2120 003752 100410
 2121 003754 012700 000001
 2122 003760 104010
 2123 003762 005337 000664
 2124 003766 001367
 2125 003770 104002
 2126 003772 000755
 2127 003774 000002
 2128
 2129
 2130
 2131
 2132
 2133
 2134
 2135
 2136
 2137 003776 013700 000602
 2138 004002 010037 000612
 2139 004006 005720
 2140 004010 010037 000614
 2141 004014 005720
 2142 004016 013737 000616 004070
 2143 004024 010037 000616
 2144 004030 005720
 2145 004032 013737 000620 004072
 2146 004040 010037 000620
 2147 004044 013737 000604 000622
 2148 004052 013737 000604 000626
 2149 004060 062737 000004 000626
 2150 004066 000207
 2151
 2152 004070 000000
 2153 004072 000000

```

:*****
:AREAD--A ROUTINE WHICH, THROUGH THE FACILITY OF
: THE MAINTENANCE BIT, OUTPUTS TO THE
: PRINTER BUFFER AND READS THE KEYBOARD
: STATUS DONE. IF THE DONE IS NOT SET
: WITHIN 600 MSEC, THE CPU WILL HALT WITH
: THE LOCATION OF THE ERROR IN RO. PRESS
: CONTINUE TO CONTINUE WITH TESTS.
:*****
  
```

```

$AREAD: MOV #600, BRCTR ;SET UP 600 MSEC DELAY
        BIS #4, @TPS ;SET MAINTENANCE BIT
        CLR @TPB ;LOAD PRINTER BUFFER
1$:     TSTB @TKS ;CHECK DONE BIT
        BMI 2$ ;BRANCH IF DONE
        MOV #1, RO ;ONE TO RO
        DELAY ;DELAY 1 MSEC.
        DEC BRCTR ;600 MSEC OVER
        BNE 1$ ;BRANCH IF NO
        EHALT
2$:     BR $AREAD ;TRY AGAIN
        RTI ;RETURN TO TEST
  
```

```

:*****
:CONIT--THIS ROUTINE SETS UP THE DEVICE ADDRESSES
: AND INTERRUPT VECTORS FOR THE CONSOLE
: TERMINAL.
:*****
  
```

```

CONIT: MOV CONADD, RO ;CONSOLE KEYBOARD STATUS ADDR TO RO
CONSET: MOV RO, TKS ;KEYBOARD STATUS ADDRESS (777560) TO TKS
        TST (RO)+ ;INCREMENT RO BY TWO
        MOV RO, TKB ;KEYBOARD DATA ADDR (777562) TO TKB
        TST (RO)+ ;INCREMENT RO BY TWO
        MOV TPS, TPSS ;SAVE TPS OF LAST TERMINAL
        MOV RO, TPS ;PRINTER STATUS ADDR(777564) TO TPS
        TST (RO)+ ;INCREMENT RO BY TWO
        MOV TPB, TPBS ;SAVE TPB OF LAST TERMINAL
        MOV RO, TPB ;PRINTER DATA ADDR (777566) TO TPB
        MOV CONVEC, TKVTR ;KEYBOARD INTERRUPT VECTOR (60) TO TKVTR
        MOV CONVEC, TPVTR ;KEYBOARD INTERRUPT VECTOR (64) TO TPVTR
        ADD #4, TPVTR
        RTS PC ;PRINTER INTERRUPT VECTOR (64) TO TPVTR

TPSS: .WORD 0 ;LAST TERM STATUS REG ADR
TPBS: .WORD 0 ;LAST TERM BUFFER REG ADR
  
```

```

2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165 004074 010237 004160
2166 004100 006302
2167 004102 062702 004166
2168
2169 004106 014237 004164
2170 004112 005037 004162
2171 004116 163701 004164
2172 004122 103403
2173 004124 005237 004162
2174 004130 000772
2175 004132 063701 004164
2176 004136 062737 000060 004162
2177 004144 113720 004162
2178 004150 005337 004160
2179 004154 001354
2180 004156 000002
2181 004160 000000
2182 004162 000000
2183 004164 000000
2184 004166 000001 000012 000144
2185 004174 001750 023420

```

```

*****
: 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
*****

$BTASC: MOV     R2,CNVCTR      ;SAVE TEN POWER
        ASL     R2            ;R2*2
        ADD     #ADTENP,R2    ;CALCULATE ADDRESS OF
                                ;STARTING TEN POWER
1$:     MOV     -(R2),TENPWR   ;POWER OF TEN VALUE TO TEN PWR
        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$:     ADD     TENPWR,R1      ;RESTORE SUBTRACTED VALUE
        ADD     #60,DIGIT     ;CONVERT (DIGIT) TO ASCII
        MOVB   DIGIT,(R0)+    ;PUT ASCII CHAR INTO USER BUFFER
        DEC     CNVCTR        ;FINISHED ALL CHARS. CALLED FOR
        BNE    1$           ;BRANCH IF NOT FINISHED
        RTI                    ;YES, EXIT
CNVCTR: .WORD   0            ;CONVERSION CHARACTER COUNT
DIGIT:  .WORD   0            ;CONVERTED CHARACTER
TENPWR: .WORD   0            ;CURRENT TEN POWER
ADTENP: .WORD   1.,10.,100.,1000.,10000.

```



```

2186
2187
2188
2189
2190
2191
2192
2193 004200 004737 003776
2194 004204 005737 000656
2195 004210 001430
2196 004212 013737 000656 000666
2197 004220 013737 000632 000670
2198 004226 105777 174436
2199 004232 100005
2200 004234 013700 000670
2201 004240 004737 004002
2202 004244 000415
2203 004246 005337 000666
2204 004252 001404
2205 004254 062737 000010 000670
2206 004262 000761
2207 004264 105777 174322
2208 004270 100350
2209 004272 105777 174314
2210 004276 100375
2211 004300 117737 174310 000700
2212 004306 113737 000700 000704
2213 004314 042737 177400 000704
2214 004322 113737 000700 000703
2215 004330 042737 177600 000700
2216 004336 023727 000700 000004
2217 004344 001715
2218 004346 012700 000011
2219 004352 042737 000377 000702
2220 004360 005300
2221 004362 001406
2222 004364 106337 000703
2223 004370 103373
2224 004372 105137 000702
2225 004376 000770
2226 004400 000002
2227
2228
2229
2230
2231
2232
2233
2234 004402 105777 174210
2235 004406 100375
2236 004410 000002

;XXXXXXXXXX
;
;READ-- A COMMON ROUTINE WHICH CHECKS THE KEYBOARD
;      DONE FLAG & SETS A FLAG INDICATING CHAR PARITY
;
;XXXXXXXXXX
$READ: JSR PC,CONIT ;RESET CONSOLE ADR AND VECTORS
        TST DLCNT ;CHECK IF MULTI DL11'S AVAILABLE
        BEQ $READC ;NONE, WAIT FOR CONSOLE INPUT
1$: MOV DLCNT,COUNT3 ;SET DL11 COUNT
    MOV FSTDL,XCSR ;ADDRESS OF FIRST DL'1 INTO XCSR
2$: TSTB @XCSR ;TEST IF ANY INPUT
    BPL 3$ ;CONTINUE IF NO INPUT
    MOV XCSR,RO ;SET THIS DL11 AS CONSOLE
    JSR PC,CONSET
    BR READ1 ;READ CHAR AND RETURN
3$: DEC COUNT3 ;DECREMENT DL11 COUNT
    BEQ 4$ ;TEST CONSOLE WHEN DONE DL11'S
    ADD #10,XCSR ;NEXT DL11 ADDRESS
    BR 2$ ;CONTINUE
4$: TSTB @TKS ;CHECK CONSOLE
    BPL 1$ ;WAIT, NO INPUT
$READC: TSTB @TKS ;CHECK KEYBOARD DONE FLAG
        BPL $READC ;BRANCH IF NOT SET
READ1: MOV @TKB,TEMPCH ;SAVE CHARACTER
        MOVB TEMPCH,PCHAR ;SAVE CODE WITH PARITY BIT
        BIC #177400,PCHAR ;MASK UNWANTED BITS
        MOVB TEMPCH,PARITY+1 ;SAVE CHAR WITH PARITY BIT
        BIC #177600,TEMPCH ;MAKE IT 7 BIT ASCII
        CMP TEMPCH,#4 ;DISREGARD EOT
        BEQ $READ
        MOV #11,RO ;SET SHIFT COUNT
        BIC #377,PARITY ;CLEAR PARITY FLAG
1$: DEC RO ;DECREMENT SHIFT COUNT
    BEQ 2$ ;EXIT IF DONE
    ASLB PARITY+1 ;SHIFT CODE
    BCC 1$ ;CONTINUE IF BIT WAS ZERO
    COMB PARITY ;CHANGE PARITY FLAG IF BIT WAS ONE
    BR 1$ ;CONTINUE
2$: RTI ;SET, RET. TO CALLER

;XXXXXXXXXX
;
;PRINT-- A COMMON ROUTINE TO CHECK THE PRINTER READY FLAG
;
;XXXXXXXXXX
$PRNT: TSTB @TPS ;CHECK PRINTER READY FLAG
        BPL $PRNT ;BRANCH IF NOT SET
        RTI ;SET, RETURN

```

2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248 004412 013737 000602 000712
2249 004420 062737 000004 000712
2250 004426 105777 174260
2251 004432 100375
2252 004434 062737 000002 000712
2253 004442 010077 174244
2254 004446 032777 020000 174240
2255 004454 001003
2256 004456 005737 000656
2257 004462 001002
2258 004464 000137 005122
2259 004470 013737 000656 000666
2260 004476 013737 000632 000670
2261 004504 005737 000710
2262 004510 001140
2263 004512 023727 000636 000020
2264 004520 002004
2265 004522 022737 104011 001772
2266 004530 001130
2267 004532 105777 174132
2268 004536 100125
2269 004540 062737 000002 000670
2270 004546 017737 174116 000700
2271 004554 042737 177600 000700
2272 004562 023727 000700 000003
2273 004570 001006
2274 004572 023727 000636 000024
2275 004600 001002
2276 004602 000137 005226
2277 004606 023727 000700 000177
2278 004614 001427
2279 004616 023727 000636 000017
2280 004624 001003
2281 004626 013703 000700
2282 004632 000461
2283 004634 023727 000636 000021
2284 004642 001004
2285 004644 013737 000700 000662
2286 004652 000451
2287 004654 023727 000636 000022
2288 004662 001056
2289 004664 013737 000700 000662
2290 004672 000441
2291 004674 023727 000636 000021
2292 004702 001011

```

*****
:PRINTC--SENDS A CHARACTER AT A TIME FIRST TO THE
:CONSOLE DL11 THEN TO ALL MULTIPLE DL11S IF
:SR BIT 13 IS = 0. IF THE REFERENCED PRINTER
:READY BIT IS NOT SET, THE CHARACTER WILL NOT BE
:SENT TO THAT PRINTER. ENTER WITH CHARACTER IN RO.
:CALL: PRINTC
*****
$PRTC: MOV CONADD,TEMP ;SET CONSOLE ADR
ADD #4,TEMP
1$: TSTB @TEMP
BPL 1$ ;WAIT FOR CONSOLE READY
ADD #2,TEMP ;SET ADR
MOV RO,@TEMP ;LOAD CONSOLE PRINTER BUFFER
BIT #BIT13,@SR ;CHECK SW 13
BNE 2$ ;SEND ALL TERMS IF SW13 DOWN
TST DLCNT ;CHECK IF MULTIPLE DL11'S
BNE 3$ ;CHECK FOR INPUT IF THERE
2$: JMP 18$
3$: MOV DLCNT,COUNT3 ;PUT NO. DL11'S INTO COUNT3
MOV FSTDL,XCSR ;ADDR OF FIRST DL INTO XCSR
4$: TST INCHK ;CHECK FOR INPUT?
BNE 13$
CMP RTNNO,#20 ;PRINTING TEST?
BGE 5$ ;BRANCH IF NOT
CMP #TTYCTL,WAITF ;KEYBOARD CONTROL?
BNE 13$ ;SKIP INPUT CHECK IF NOT
5$: TSTB @XCSR ;TEST IF ANY INPUT
BPL 13$ ;CONTINUE IF NO INPUT
ADD #2,XCSR ;SET BUFFER ADDRESS
MOV @XCSR,TEMPCH
BIC #177600,TEMPCH
CMP TEMPCH,#3 ;CHECK IF CONTROL-C
BNE 6$ ;CONTINUE IF NOT
CMP RTNNO,#24 ;CHECK IF TEST 24
BNE 6$ ;CONTINUE IF NOT CONTROL-C
6$: JMP 20$
CMP TEMPCH,#177 ;CHECK IF RUBOUT
BEQ 9$ ;YES, CHECK TEST NUMBER
CMP RTNNO,#17 ;TEST 17?
BNE 7$ ;BRANCH IF NOT
MOV TEMPCH,R3 ;SAVE CHARACTER
BR 12$ ;CONTINUE
7$: CMP RTNNO,#21 ;TEST 21?
BNE 8$ ;BRANCH IF NOT
MOV TEMPCH,REPT ;SAVE CHARACTER
BR 12$ ;CONTINUE
8$: CMP RTNNO,#22 ;TEST 22?
BNE 14$ ;CONTINUE IF NOT
MOV TEMPCH,REPT ;SAVE CHARACTER
BR 12$ ;CONTINUE
9$: CMP RTNNO,#21 ;CHECK IF TEST 21
BNE 10$ ;NO, CHECK IF TEST 22

```


2349	005226	012700	000036			20\$:	MOV #30.,R0	:DELAY FOR HALF DUPLEX
2350	005232	104010					DELAY	
2351	005234	104012					CRLF	:SEND CR-LF
2352	005236	022626					POPSP2	:RESET STACK
2353	005240	000137	013030				JMP E024B	:RETURN TO TEST
2354	005244	023727	000700	000177	21\$:	CMP TEMPCH,#177	:CHECK IF RUBOUT	
2355	005252	001006				BNE 23\$:BRANCH IF NO	
2356	005254	000607				BR 9\$		
2357	005256	012737	000001	000634	22\$:	MOV #1,CNTLSW	:CLEAR LOOP AND SEQUENCE BITS	
2358	005264	000137	002154			JMP TTY1B	:GO WAIT FOR NEXT TEST	
2359	005270	010046				23\$:	MOV RO,-(SP)	:SAVE RO
2360	005272	012700	000036			MOV #30.,R0	:DELAY FOR HALF DUPLEX	
2361	005276	104010				DELAY		
2362	005300	012600				MOV (SP)+,R0	:RESTORE RO	
2363	005302	023727	000636	000017		CMP RTNNO,#17	:CHECK IF TEST 17	
2364	005310	001002				BNE 24\$:BRANCH IF NOT TEST 17	
2365	005312	013703	000700			MOV TEMPCH,R3	:STORE INPUTTED CHARACTER	
2366	005316	023727	000636	000021	24\$:	CMP RTNNO,#21	:CHECK IF TEST 21	
2367	005324	001003				BNE 25\$:BRANCH IF NOT TEST 21	
2368	005326	013737	000700	000662		MOV TEMPCH,REPT	:STORE INPUTTED CHARACTER	
2369	005334	023727	000636	000022	25\$:	CMP RTNNO,#22	:CHECK IF TEST 22	
2370	005342	001003				BNE 26\$:BRANCH IF NOT TEST 22	
2371	005344	013737	000700	000662		MOV TEMPCH,REPT	:STORE INPUTTED CHARACTER	
2372	005352	000002			26\$:	RTI	:RETURN TO TEST	
2373								

CZLACFO LA36 TERM (DL11 & KL11)
CZLACF.P11 13-JUN-79 15:33

MACY11 30A(1052) 13-JUN-79 15:35 PAGE 55
COMMON ROUTINES USED BY LA36 TESTS

SEQ 0055

2374

.SBTTL I/O LOGIC TESTS

2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421

005354 000040
005356 005406
005360 000012
005362 005372
005364 012737 005402 000004
005372 005777 173214
005376 104005
005400 000774
005402 104001
005404 000774

005406 000041
005410 005440
005412 000012
005414 005424
005416 012737 005434 000004
005424 005777 173164
005430 104005
005432 000774
005434 104001
005436 000774

: ONLY THE CONSOLE TERMINAL IS TESTED.
: UPON COMPLETION, THE CPU WILL EITHER HALT IF SR
: BIT8 IS = 1 AND AWAIT FUTHER INSTRUCTIONS OR CONTINUE
: AND EXECUTE THE PRINTER TESTS CONTINUOUSLY
: IF AN I/O TEST FAILS, THE CPU WILL HALT AT ERRHLT
: WITH THE ADDRESS OF THE ERROR IN RO (LOC 777700). PRESSING
: THE CONTINUE SWITCH WILL CAUSE THE I/O TEST TO
: CONTINUE WITH THE NEXT TEST. HOWEVER IF SWITCH 14
: WERE SET, OR IS SET BEFORE THE CONTINUE SWITCH IS
: PRESSED, THE FAILED TEST WILL LOOP ON ITSELF
: WITHOUT FURTHER HALTS

: ATO-- TEST #40--TESTS THE ABILITY TO REFERENCE THE
: RECEIVER STATUS WORD (TKS) WITHOUT TRAPPING.
:*****

ATO: 40 ;TEST NUMBER
ATOX: AT1 ;NEXT TEST
10. ;ITERATION COUNT
1\$;SCOPE ENTRY
MOV #3\$,MACHER ;SET UP MACHINE ERROR TRAP
1\$: TST @TKS ;REFERENCE RECEIVER STATUS WORD
2\$: CHAIN ;CHAIN TO NEXT TEST
BR 1\$;REPEAT TEST
3\$: ERROR ;ERROR TRAPPED WHEN REFERENCING
BR 2\$;RECEIVER STATUS WORD (TKS)

: AT1--TEST #41--TESTS THE ABILITY TO REFERENCE THE
: RECEIVER BUFFER (TKB) WITHOUT TRAPPING.
:*****

AT1: 41 ;TEST NUMBER
AT2 ;NEXT TEST
10. ;ITERATION COUNT
1\$;SCOPE ENTRY
MOV #3\$,MACHER ;SET UP MACHINE ERROR TRAP
1\$: TST @TKB ;REFERENCE RECEIVER BUFFER
2\$: CHAIN ;CHAIN TO NEXT TEST
BR 1\$;REPEAT TEST
3\$: ERROR ;TRAPPED WHEN REFERENCING
BR 2\$;RECEIVER BUFFER (TKB)

2422
2423
2424
2425
2426
2427 005440 000042
2428 005442 005472
2429 005444 000012
2430 005446 005456
2431 005450 012737 005466 000004
2432 005456 005777 173134
2433 005462 104005
2434 005464 000774
2435 005466 104001
2436 005470 000774
2437
2438
2439
2440
2441
2442
2443 005472 000043
2444 005474 005524
2445 005476 000012
2446 005500 005510
2447 005502 012737 005520 000004
2448 005510 005777 173104
2449 005514 104005
2450 005516 000774
2451 005520 104001
2452 005522 000774

```
*****  
:AT2--TEST #42--TESTS THE ABILITY TO REFERENCE THE  
: TRANSMITTER STATUS WORD (TPS) WITHOUT TRAPPING.  
*****  
AT2: 42 :TEST NUMBER  
AT3 :NEXT TEST  
10. :ITERATION COUNT  
1$ :SCOPE ENTRY  
MOV #3$,MACHER :SET UP MACHINE ERROR TRAP  
1$: TST @TPS :REFERENCE TRANSMITTER STATUS  
2$: CHAIN :CHAIN TO NEXT TEST  
BR 1$ :REPEAT TEST  
3$: ERROR :TRAPPED WHEN REFERENCING  
BR 2$ :TRANSMITTER STATUS WORD
```

```
*****  
:AT3-- TEST #43--TESTS THE ABILITY TO REFERENCE THE  
: TRANSMITTER BUFFER (TPB) WITHOUT TRAPPING.  
*****  
AT3: 43 :TEST NUMBER  
AT4 :NEXT TEST  
10. :ITERATION COUNT  
1$ :SCOPE ENTRY  
MOV #3$,MACHER :SET UP ERROR TRAP  
1$: TST @TPB :REFERENCE TRANSMITTER BUFFER  
2$: CHAIN :CHAIN TO NEXT TEST  
BR 1$ :REPEAT TEST  
3$: ERROR :TRAPPED WHEN REFERENCING  
BR 2$ :TRANSMITTER BUFFER.
```

2453
2454
2455
2456
2457
2458 005524 000044
2459 005526 005614
2460 005530 000012
2461 005532 005546
2462 005534 012746 000340
2463 005540 012746 005546
2464 005544 000002
2465 005546 052777 000100 173036
2466 005554 032777 000100 173030
2467 005562 001002
2468 005564 104001
2469 005566 000410
2470 005570 042777 000100 173014
2471 005576 032777 000100 173006
2472 005604 001401
2473 005606 104001
2474 005610 104005
2475 005612 000755
2476
2477
2478
2479
2480
2481
2482 005614 000045
2483 005616 005672
2484 005620 000012
2485 005622 005636
2486 005624 012746 000340
2487 005630 012746 005636
2488 005634 000002
2489 005636 052777 000100 172746
2490 005644 105777 172746
2491 005650 001775
2492 005652 000005
2493 005654 032777 000100 172730
2494 005662 001401
2495 005664 104001
2496 005666 104005
2497 005670 000762

:AT4-- TEST #44--TESTS THE ABILITY TO SET AND CLEAR THE
: RECEIVER INTERRUPT ENABLE BIT.
:*****

AT4: 44 ;TEST NUMBER
AT5 ;NEXT TEST
10. ;ITERATION COUNT
1\$;SCOPE ENTRY
MOV #PRTY7,-(SP) ;SET PRIORITY 7
MOV #1\$,-(SP)
RTI
1\$: BIS #BIT6,@TKS ;SET INTERRUPT ENABLE BIT
BIT #BIT6,@TKS ;CHECK IF BIT IS SET
BNE 3\$;BRANCH IF SET
2\$: ERROR ;NOT SET, ERROR
BR 5\$;CHAIN TO NEXT TEST
3\$: BIC #BIT6,@TKS ;CLEAR INTERRUPT ENABLE BIT
BIT #BIT6,@TKS ;CHECK IF BIT IS CLEARED
BEQ 5\$;BRANCH IF CLEARED
4\$: ERROR ;NOT CLEARED, ERROR
5\$: CHAIN ;CHAIN TO NEXT TEST
BR 1\$;DO TEST AGAIN

:AT5-- TEST #45--CHECKS THAT THE RECEIVER INTERRUPT
: ENABLE BIT CAN BE CLEARED WITH RESET INSTRUCTION.
:*****

AT5: 45 ;TEST NUMBER
AT6 ;NEXT TEST
10. ;ITERATION COUNT
1\$;SCOPE ENTRY
MOV #PRTY7,-(SP) ;SET PRIORITY TO 7
MOV #1\$,-(SP)
RTI
1\$: BIS #BIT6,@TKS ;SET INTERRUPT ENABLE BIT
3\$: TSTB @TPS ;BE SURE PRINTER IS DONE WITH DL11S1 MESSAGE
BEQ 3\$;BEFORE ALLOWING FOLLOWING RESET.
RESET ;RESET
2493 BIT #BIT6,@TKS ;TEST INTERRUPT ENABLE BIT
BEQ 2\$;BRANCH IF CLEARED
2\$: ERROR ;STILL SET,ERROR
CHAIN ;CHAIN TO NEXT ROUTINE
BR 1\$;REPEAT TEST

2498
2499
2500
2501
2502
2503 005672 000046
2504 005674 005762
2505 005676 000012
2506 005700 005714
2507 005702 012746 000340
2508 005706 012746 005714
2509 005712 000002
2510 005714 052777 000100 172674
2511 005722 032777 000100 172666
2512 005730 001002
2513 005732 104001
2514 005734 000410
2515 005736 042777 000100 172652
2516 005744 032777 000100 172644
2517 005752 001401
2518 005754 104001
2519 005756 104005
2520 005760 000755
2521
2522
2523
2524
2525
2526
2527 005762 000047
2528 005764 006032
2529 005766 000012
2530 005770 006004
2531 005772 012746 000340
2532 005776 012746 006004
2533 006002 000002
2534 006004 052777 000100 172604
2535 006012 000005
2536 006014 032777 000100 172574
2537 006022 001401
2538 006024 104001
2539 006026 104005
2540 006030 000765

:AT6-- TEST#46--TESTS THE ABILITY TO SET AND CLEAR
: TRANSMITTER INTERRUPT ENABLE BIT.

AT6: 46 ;TEST NUMBER
AT7 ;NEXT TEST
10. ;ITERATION COUNT
1\$;SCOPE ENTRY
MOV #PRTY7,-(SP) ;SET PRIORITY TO 7
MOV #1\$,-(SP)
RTI
1\$: BIS #BIT6,@TPS ;SET INTERRUPT ENABLE BIT
BIT #BIT6,@TPS ;CHECK THAT BIT IS SET
BNE 2\$;BRANCH IF SET
ERROR ;NOT SET, ERROR
BR 3\$;CHAIN TO NEXT TEST
2\$: BIC #BIT6,@TPS ;CLEAR INTERRUPT ENABLE BIT
BIT #BIT6,@TPS ;CHECK IF BIT IS CLEARED
BEQ 3\$;BRANCH IF CLEARED
ERROR ;NOT CLEARED, ERROR
3\$: CHAIN ;CHAIN TO NEXT TEST
BR 1\$;DO AGAIN

:AT7-- TEST #47--TESTS THE ABILITY TO CLEAR TRANSMITTER
: INTERRUPT ENABLE BIT WITH RESF1 INSTRUCTION.

AT7: 47 ;TEST NUMBER
AT10 ;NEXT TEST
10. ;ITERATION COUNT
1\$;SCOPE ENTRY
MOV #PRTY7,-(SP) ;SET PRIORITY TO 7
MOV #1\$,-(SP)
RTI
1\$: BIS #BIT6,@TPS ;SET INTERRUPT BIT
RESET ;RESET
BIT #BIT6,@TPS ;CHECK IF BIT IS CLEARED
BEQ 2\$;BRANCH IF CLEARED
ERROR ;ERROR, RESET DID NOT CLEAR BIT
2\$: CHAIN ;CHAIN TO NEXT ROUTINE
BR 1\$;REPEAT TEST

```
2541
2542
2543
2544
2545
2546 006032 000050
2547 006034 006070
2548 006036 000012
2549 006040 006042
2550 006042 032777 001000 172644 1$:
2551 006050 001005
2552 006052 000005
2553 006054 105777 172536
2554 006060 100401
2555 006062 104001
2556 006064 104005
2557 006066 000765
2558
2559
2560
2561
2562
2563
2564 006070 000051
2565 006072 006130
2566 006074 000012
2567 006076 006100
2568 006100 012700 000226
2569 006104 104010
2570 006106 000005
2571 006110 005077 172504
2572 006114 105777 172476
2573 006120 100001
2574 006122 104001
2575 006124 104005
2576 006126 000764

:*****
:AT10-- TEST #50--CHECKS THAT RESET SETS THE TRANSMITTER
:         READY BIT AND THAT THE READY BIT CAN BE READ RELIABLY.
:*****
AT10: 50           ;TEST NUMBER
      AT11        ;NEXT TEST
      10.         ;ITERATION COUNT
      1$         ;SCOPE ENTRY
      BIT #LSI11,@SR ;SKIP TEST IF AN LSI-11
      BNE 2$
      RESET      ;RESET
      TSTB @TPS  ;CHECK TRANSMIT READY BIT
      BMI 2$    ;BRANCH IF SET
      ERROR    ;ERROR, RESET DID NOT SET READY BIT
2$:   CHAIN      ;CHAIN TO NEXT TEST
      BR 1$     ;DO AGAIN

:*****
:AT11-- TEST #51--TESTS THAT THE TRANSMITTER READY RESETS
:         BY LOADING THE TRANSMITTER BUFFER.
:*****
AT11: 51           ;TEST NUMBER
      AT12        ;NEXT TEST
      10.         ;ITERATION COUNT
      1$         ;SCOPE ENTRY
      MOV #226,R0 ;DELAY 150 MSEC.
      DELAY      ;RESET
      RESET     ;LOAD TRANSMITTER BUFFER
      CLR @TPB  ;CHECK TRANSMIT READY BIT
      TSTB @TPS ;BRANCH IF CLEARED
      BPL 2$    ;NOT CLEARED, ERROR
      ERROR    ;CHAIN TO NEXT TEST
2$:   CHAIN      ;REPEAT TEST
      BR 1$
```

2577
2578
2579
2580
2581
2582 006130 000052
2583 006132 006204
2584 006134 000012
2585 006136 006144
2586 006140 104004
2587 006142 006200
2588 006144 000005
2589 006146 005077 172444
2590 006152 005046
2591 006154 012746 006162
2592 006160 000002
2593 006162 052777 000100 172426
2594 006170 000240
2595 006172 104001
2596 006174 104005
2597 006176 000762
2598 006200 022626
2599 006202 000774
2600
2601
2602
2603
2604
2605
2606 006204 000035
2607 006206 006264
2608 006210 000012
2609 006212 006220
2610 006214 104004
2611 006216 006256
2612 006220 013746 000630
2613 006224 012746 006232
2614 006230 000002
2615 006232 005077 172360
2616 006236 052777 000100 172352
2617 006244 000240
2618 006246 005077 172344
2619 006252 104005
2620 006254 000761
2621 006256 022626
2622 006260 104001
2623 006262 000771

```
*****  
:AT12-- TEST #52--CHECKS THAT THE TRANSMIT READY BIT CAN  
: CAUSE AN INTERRUPT  
*****  
AT12: 52 ;TEST NUMBER  
AT13 ;NEXT TEST  
10. ;ITERATION COUNT  
1$ ;SCOPE ENTRY  
STPCHV ;SET UP TRANSMITTER INTERRUPT VECTOR  
4$ ;TO 4$  
1$: RESET ;SEE CHAINY COMMENT  
CLR @TPS ;DISABLE TRANSMIT INTERRUPT  
CLR -(SP) ;SET PRIORITY TO ZERO  
MOV #2$,-(SP)  
RTI  
2$: BIS #BIT6,@TPS ;ENABLE TRANSMIT INTERRUPT  
NOP  
ERROR ;TRANSMIT READY DID NOT CAUSE INTERRUPT  
3$: CHAIN ;CHAIN TO NEXT TEST  
BR 1$ ;REPEAT TEST  
4$: POPSP2 ;INTERRUPT OCCURRED, CLEAN STACK  
BR 3$ ;CHAIN TO NEXT TEST
```

```
*****  
:AT13-- TEST#53--TESTS THAT THE TRANSMIT READY DOES NOT CAUSE AN  
: INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL  
*****  
AT13: 35 ;TEST NUMBER  
AT14 ;NEXT TEST  
10. ;ITERATION COUNT  
1$ ;SCOPE ENTRY  
STPCHV ;SET UP TRANSMIT INTERRUPT  
4$ ;VECTOR TO 4$  
1$: MOV TPLVL,-(SP) ;SET PROCESSOR TO SAME LEVEL AS XMITTER  
MOV #2$,-(SP)  
RTI  
2$: CLR @TPS ;DISABLE TRANSMITTER INTERRUPTS  
BIS #BIT6,@TPS ;ENABLE TRANSMITTER INTERRUPTS  
NOP  
3$: CLR @TPS ;OK, NO INTERRUPT OCCURRED  
CHAIN ;CHAIN TO NEXT TEST  
BR 1$ ;REPEAT TEST  
4$: POPSP2 ;INTERRUPT OCCURRED,ERROR,CLEAN  
ERROR ;UP STACK  
BR 3$ ;CHAIN TO NEXT TEST
```

2624
2625
2626
2627
2628
2629
2630 006264 000054
2631 006266 006350
2632 006270 000012
2633 006272 006300
2634 006274 104004
2635 006276 006336
2636 006300 005077 172312
2637 006304 013746 000630
2638 006310 162716 000040
2639 006314 012746 006322
2640 006320 000002
2641 006322 052777 000100 172266 28:
2642 006330 000240
2643 006332 104001
2644 006334 000401
2645 006336 022626
2646 006340 005077 172252
2647 006344 104005
2648 006346 000754

```
*****  
:AT14-- TEST#54--TESTS THAT THE TRANSMIT READY DOES CAUSE AN  
: INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY LEVEL  
: ONE LOWER THAN THE TRANSMIT INTERRUPT REQUEST LEVEL  
:*****  
AT14: 54 ;TEST NUMBFR  
AT15 ;NEXT TEST  
10. ;ITERATION COUNT  
1$ ;SCOPE ENTRY  
STPCHV ;SET UP TRANSMIT INTERRUPT  
3$ ;VECTOR TO 3$  
1$: CLR @TPS ;DISABLE TRANSMIT INTERRUPTS  
MOV TPLVL,-(SP) ;SET PROCESSOR PRIORITY ONE  
SUB #40,(SP) ;LEVEL LOWER THAN TRANSMITTER  
MOV #2$,-(SP)  
RTI  
2$: BIS #BIT6,@TPS ;ENABLE TRANSMITTER INTERRUPTS  
NOP  
ERROR ;NO INTERRUPT, ERROR  
BR 4$ ;CHAIN TO NEXT TEST  
3$: POPSP2 ;INTERRUPT OCCURED, OK, CLEAN STACK  
4$: CLR @TPS ;DISABLE TRANSMITTER INTERRUPTS  
CHAIN ;CHAIN TO NEXT TEST  
BR 1$ ;REPEAT TEST
```

```

2649
2650
2651
2652
2653
2654
2655 006350 000055
2656 006352 006450
2657 006354 000012
2658 006356 006360
2659 006360 104004
2660 006362 006422
2661 006364 005077 172226
2662 006370 005046
2663 006372 012746 006400
2664 006376 000002
2665 006400 052777 000100 172210 2$:
2666 006406 000240
2667 006410 104001
2668 006412 005077 172200 3$:
2669 006416 104005
2670 006420 000757
2671 006422 012777 006442 172176 4$:
2672 006430 012716 006436
2673 006434 000002
2674 006436 000240 5$:
2675 006440 000764
2676 006442 022626 6$:
2677 006444 104001
2678 006446 000761
2679
2680
2681
2682
2683
2684 006450 000056
2685 006452 006516
2686 006454 000012
2687 006456 006460
2688 006460 032777 001000 172226 1$:
2689 006466 001011
2690 006470 012700 000226
2691 006474 104010
2692 006476 104021
2693 006500 000005
2694 006502 105777 172104
2695 006506 100001
2696 006510 104001
2697 006512 104005
2698 006514 000761

```

```

:*****
:AT15-- TEST#55--TESTS THAT THE TRANSMIT READY DOES NOT
: REINTERRUPT AFTER AN RTI WHEN THE READY BIT HAS
: NOT BEEN RESET.
:*****
AT15: 55 ;TEST NUMBER
      10. ;NEXT TEST
      1$ ;ITERATION COUNT
1$: STPCHV ;SCOPE ENTRY
     4$ ;SET TRANSMIT INTERRUPT VECTOR
     CLR @TPS ;DISABLE TRANSMITTER INTERRUPTS
     CLR -(SP) ;SET PROCESSOR PRIORITY TO ZERO
     MOV #2$,-(SP)
     RTI
2$: BIS #B116,@TPS ;ENABLE TRANSMITTER INTERRUPTS
     NOP
     ERROR ;ERROR1, TRANSMITTER FAILED TO INTERRUPT
3$: CLR @TPS ;DISABLE TRANSMITTER INTERRUPTS
     CHAIN ;CHAIN TO NEXT TEST
     BR 1$ ;REPEAT TEST
4$: MOV #6$,@TPVTR ;INTERRUPT OCCURRED, CHANGE INTERRUPT
     MOV #5$,@SP ;VECTOR TO 6$ AND RETURN TO 5$
     RTI ;RETURN FROM INTERRUPT
5$: NOP
     BR 3$ ;CHAIN TO NEXT TEST
6$: POPSP2 ;ERROR2, TRANSMITTER REINTERRUPTED
     ERROR ;AFTER RTI WITH READY BIT LEFT ON.
     BR 3$ ;CLEAN STACK, CHAIN TO NEXT TEST.
:*****
:AT16--TEST#56--CHECKS THAT RESET CLEARS THE RECEIVER DONE BIT
:*****
AT16: 56 ;TEST NUMBER
      10. ;NEXT TEST
      1$ ;ITERATION COUNT
1$: BIT #LSI11,@SR ;SKIP TEST IF LSI-11
     BNE 3$
     MOV #226,R0
2$: DELAY ;DELAY 150 MSEC.
     AREAD ;ENABLE RECEIVER
     RESET ;RESET
     TSTB @TKS ;TEST DONE BIT
     BPL 3$ ;BRANCH IF DONE IS CLEARED
3$: ERROR ;NOT CLEARED, ERROR
     CHAIN ;CHAIN TO NEXT TEST
     BR 1$ ;REPEAT TEST

```

```

2699
2700
2701
2702
2703
2704 006516 000057
2705 006520 006566
2706 006522 000012
2707 006524 006526
2708 006526 032777 001000 172160 1$:
2709 006534 001012
2710 006536 012700 000226
2711 006542 104010
2712 006544 104021
2713 006546 105777 172042
2714 006552 105777 172034
2715 006556 100001
2716 006560 104001
2717 006562 104005
2718 006564 000760
2719
2720
2721
2722
2723
2724
2725 006566 000060
2726 006570 006660
2727 006572 000012
2728 006574 006602
2729 006576 104003
2730 006600 006652
2731 006602 032777 001000 172104 1$:
2732 006610 001021
2733 006612 012700 000226
2734 006616 104010
2735 006620 104021
2736 006622 005077 171764 2$:
2737 006626 005046
2738 006630 012746 006636
2739 006634 000002
2740 006636 052777 000100 171746 3$:
2741 006644 000240
2742 006646 104001
2743 006650 000401
2744 006652 022626
2745 006654 104005
2746 006656 000751

```

```

:*****
:AT17-- TEST#57--CHECKS THAT REFERENCING THE RECEIVER BUFFER
:      CLEARS THE DONE BIT.
:*****
AT17:  57          ;TEST NUMBER
        AT20       ;NEXT TEST
        10.        ;ITERATION COUNT
        1$        ;SCOPE ENTRY
        BIT        #LSI11,@SR ;CHECK FOR LSI-11
        BNF        3$ ;SKIP TEST IF SET
        MOV        #226,R0
        DELAY      ;DELAY 150 MSEC.
        AREAD     ;ENABLE RECEIVER
        TSTB      @TKB  ;REFERENCE RECEIVER BUFFER
        TSTB      @TKS  ;TEST DONE BIT
        BPL        3$  ;BRANCH IF NOT SET
        ERROR     ;DONE BIT IS SET, ERROR
        CHAIN     ;CHAIN TO NEXT TEST
        BR        1$  ;REPEAT TEST

:*****
:AT20-- TEST#60--CHECK THAT THE RECEIVER DONE BIT IS ABLE TO
:      CAUSE AN INTERRUPT.
:*****
AT20:  60          ;TEST NUMBER
        AT21       ;NEXT TEST
        10.        ;ITERATION COUNT
        1$        ;SCOPE ENTRY
        STRDRV    ;SET UP RECEIVER INTERRUPT
        4$        ;VECTOR TO 4$
        BIT        #LSI11,@SR ;CHECK FOR LSI-11
        BNE        5$ ;SKIP TEST IF SET
        MOV        #226,R0
        DELAY      ;DELAY 150 MSEC
        AREAD     ;ENABLE RECEIVER
        CLR        @TKS  ;DISABLE RECEIVER INTERRUPTS
        CLR        -(SP) ;SET PROCESS STATUS TO ZERO
        MOV        #3$,-(SP)
        RTI
        BIS        #BIT6,@TKS ;ENABLE RECEIVER INTERRUPT
        NOP
        ERROR     ;ERROR,RECEIVER FAILED TO INTERRUPT
        BR        5$  ;CHAIN TO NEXT TEST
        POPSP2   ;OK, CLEAN STACK
        CHAIN     ;CHAIN TO NEXT TEST
        BR        1$  ;REPEAT TEST

```

```

2747
2748
2749
2750
2751
2752
2753 006660 000061
2754 006662 006764
2755 006664 000012
2756 006666 006700
2757 006670 104003
2758 006672 006756
2759 006674 005077 171712
2760 006700 032777 001000 172006
2761 006706 001017
2762 006710 012700 000226
2763 006714 104010
2764 006716 104021
2765 006720 005077 171666
2766 006724 013746 000624
2767 006730 012746 006736
2768 006734 000002
2769 006736 052777 000100 171646
2770 006744 000240
2771 006746 005077 171640
2772 006752 104005
2773 006754 000751
2774 006756 022626
2775 006760 104001
2776 006762 000771

:*****
:AT21-- TEST#61--TESTS THAT THE RECEIVER DONE DOES NOT CAUSE AN
:          INTERRUPT WHEN THE PROCESSOR IS AT THE SAME LEVEL AS
:          THE RECEIVER'S INTERRUPT REQUEST LEVEL.
:*****

AT21: 61 ;TEST NUMBER
      AT22 ;NEXT TEST
      10. ;ITERATION COUNT
      ADD2 ;SCOPE ENTRY
      STRDRV ;SET RECEIVER VECTOR TO ADD1
      ADD1

CHGF4: CLR @TKS ;DISABLE INTERRUPTS
ADD2: BIT #LSI11,@SR ;CHECK FOR LSI-11
      BNE ADD3 ;SKIP TEST IF SET
      MOV #226,R0

      DELAY ;DELAY 150 MSEC
      AREAD ;ENABLE RECEIVER

2$: CLR @TKS ;DISABLE RECEIVER INTERRUPTS
   MOV TKLVL,-(SP) ;SET PROCESSOR PRIORITY TO SAME LEVEL AS RECEIVER
   MOV #3$,-(SP)

3$: BIS #BIT6,@TKS ;ENABLE RECEIVER INTERRUPTS
   NOP

ADD3: CLR @TKS ;OK, NO INTERRUPT OCCURRED
      CHAIN ;CHAIN TO NEXT TEST
      BR ADD2 ;REPEAT TEST
ADD1: POPSP2 ;ERROR, RECEIVER INTERRUPTED, CLEAN STACK
      ERROR
      BR ADD3 ;BRANCH ADD3

```

```
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784 006764 000062  
2785 006766 007072  
2786 006770 000012  
2787 006772 007000  
2788 006774 104003  
2789 006776 007060  
2790 007000 032777 001000 171706 1$:  
2791 007006 001025  
2792 007010 012700 000226  
2793 007014 104010  
2794 007016 104021  
2795 007020 005077 171566 2$:  
2796 007024 013746 000624  
2797 007030 012746 007036  
2798 007034 000002  
2799 007036 162737 000040 177776 3$:  
2800 007044 052777 000100 171540  
2801 007052 000240  
2802 007054 104001  
2803 007056 000401  
2804 007060 022626 4$:  
2805 007062 005077 171524 5$:  
2806 007066 104005  
2807 007070 000743
```

```
*****  
;AT22-- TEST#62--TESTS THAT THE RECEIVER DONE DOES CAUSE AN  
; INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY ONE  
; LEVEL LOWER THAN THE RECEIVER'S INTERRUPT  
; REQUEST LEVEL  
*****  
AT22: 62 ;TEST NUMBER  
AT23 ;NEXT TEST  
10. ;ITERATION COUNT  
1$ ;SCOPE ENTRY  
STRDRV ;SET RECEIVER INTERRUPT  
4$ ;VECTOR TO 4$  
BIT #LS111,DSR ;CHECK FOR LS111  
BNE 5$ ;SKIP TEST IF SET  
MOV #226,R0  
DELAY ;DELAY 150 MSEC  
AREAD ;ENABLE RECEIVER  
2$: CLR @TKS ;DISABLE READER INTERRUPTS  
MOV TKLVL,-(SP) ;SET PROCESSOR PRIORITY ONE LEVEL  
MOV #3$,-(SP)  
RTI  
3$: SUB #40,PSW ;LOWER THAN READER  
BIS #BIT6,@TKS ;ENABLE INTERRUPTS  
NOP  
ERROR ;FAILED TO INTERRUPT  
BR 5$ ;CHAIN TO NEXT TEST  
4$: POPSP2 ;OK, CLEAN STACK  
5$: CLR @TKS ;DISABLE RECEIVER INTERRUPTS  
CHAIN ;CHAIN TO NEXT TEST  
BR 1$ ;REPEAT TEST
```


2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840

007072 000063
007074 007204
007076 000012
007100 007102
007102 032777 001000 171604
007110 001015
007112 012700 000226
007116 104010
007120 104021
007122 104003
007124 007156
007126 005077 171460
007132 052777 000100 171452
007140 000240
007142 104001
007144 005077 171442
007150 000005
007152 104005
007154 000752
007156 012777 007176 171436
007164 012716 007172
007170 000002
007172 000240
007174 000763
007176 022626
007200 104001
007202 000760

```
*****  
:AT23-- TEST#63--CHECKS THAT THE RECEIVER DONE DOES NOT  
: REINTERRUPT AFTER RTI INSTRUCTION WHEN DONE  
: BIT IS LEFT SET.  
*****  
AT23: 63 ;TEST NUMBER  
AT24 ;NEXT TEST  
10. ;ITERATION COUNT  
1$ ;SCOPE ENTRY  
BIT #LSI11,@SR ;CHECK FOR LSI-11  
BNE 3$ ;SKIP TEST IF SET  
MOV #226,R0  
DELAY ;DELAY 150 MSEC  
AREAD ;ENABLE RECEIVER  
STRDRV ;SET RECEIVER INTERRUPT  
4$ ;VECTOR TO 4$  
CLR @TKS ;DISABLE RECEIVER INTERRUPTS  
BIS #BIT6,@TKS ;ENABLE RECEIVER INTERRUPT  
NOP  
ERROR ;NO INTERRUPT, ERROR  
CLR @TKS ;DISABLE RECEIVER INTERRUPTS  
RESET ;RESET AFTER LAST INTERRUPT  
CHAIN ;CHAIN TO NEXT TEST  
BR 1$ ;REPEAT TEST  
MOV #6$,@TKVTR ;INTERRUPT, OK, CHANGE VECTOR TO 6$  
MOV #5$,@SP ;CHANGE RET ADDR TO 5$  
RTI ;RETURN  
5$: NOP  
BR 3$ ;OK, NO ADDITIONAL INTERRUPT  
6$: POPSP2 ;ERROR, ADDITIONAL INTERRUPT  
ERROR  
BR 3$ ;CHAIN TO NEXT TEST
```

```

2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852 007204
2853 007204 000064
2854 007206 177777
2855 007210 000001
2856 007212 007214
2857 007214 032777 001000 171472 1$:
2858 007222 001455
2859 007224 005777 171362
2860 007230 001401
2861 007232 104001
2862 007234 012700 000600
2863 007240 012737 000030 000716 2$:
2864 007246 104000
2865 007250 015101
2866 007252 104010
2867 007254 105777 171332 3$:
2868 007260 100407
2869 007262 005337 000716
2870 007266 001403
2871 007270 012700 000600
2872 007274 000766
2873 007276 104001 4$:
2874
2875 007300 104003 5$:
2876 007302 007322
2877 007304 052777 000100 171300
2878 007312 000240
2879 007314 000240
2880 007316 104001 6$:
2881 007320 022626 7$:
2882 007322 017737 171266 000716 8$:
2883 007330 105777 171256
2884 007334 100001
2885 007336 104001
2886 007340 104003 9$:
2887 007342 007366
2888 007344 052777 000100 171240
2889 007352 000240
2890 007354 000240
2891 007356 005077 171230 10$:
2892 007362 104005
2893 007364 000713
2894 007366 104001 11$:
2895 007370 022626
2896 007372 000771
  
```

```

*****
:AT24--TEST#64--HAVE OPERATOR TYPE A CHARACTER ON THE
:      KEYBOARD, THEN CHECK FOR RECEIVER DONE.
:      CHECK THAT RECEIVER DONE CAUSES AN INTERRUPT
:      WHEN BIT 6 (INTERRUPT ENABLE) IS SET.
:      CHECK THAT READING TKB CLEARS DONE BIT AND
:      THAT DONE CLEARED DOES NOT CAUSE AN INTERRUPT.
:
:      ALLOW 12 SECONDS FOR OPERATOR RESPONSE.
*****
  
```

```

CHGF7:
AT24: 64 ;TEST NUMBER
      -1 ;LAST TEST
      1 ;ITERATION COUNT
      1$ ;SCOPE ENTRY
      BIT #LSI11,@SR ;SKIP TEST IF NOT AN LSI-11
      BEQ 10$
      TST @TKS ;SHOULD BE CLEAR
      BEQ 2$
      ERROR ;RECEIVER STATUS NOT -0
      MOV #600,R0 ;1/2 SEC DELAY
      MOV #30,CNTR ;SET UP FOR 12 SEC WAIT
      TYPE
      OPMSG ;MESSAGE TO TYPE A CHARACTER
      DELAY ;1/2 SECOND
      TSTB @TKS ;CHECK DONE BIT
      BMI 5$ ;SET - EXIT LOOP
      DEC CNTR
      BEQ 4$
      MOV #600,R0 ;TIME HAS RUN OUT...
      BR 3$ ;ANOTHER 1/2 SEC
      ERROR ;CONTINUE WAIT
      ;NO RECEIVER DONE, OR
      ;OPERATOR DID NOT RESPOND
      ;SET RECEIVER INTERRUPT
      ;TO 8$
      BIS #BIT6,@TKS ;ENABLE INTERRUPT
      NOP
      NOP
      ERROR ;RECEIVER DID NOT INTERRUPT
      POPSP2 ;CLEAN UP THE STACK
      MOV @TKB,CNTR ;READ DATA BUFFER
      TSTB @TKS ;CHECK THE DONE BIT
      BPL 9$ ;OK
      ERROR ;READING DATA BUFFER DID NOT CLEAR DONE
      STRDRV ;SET RECEIVER INTERRUPT
      11$ ;VECTOR TO 11$
      BIS #BIT6,@TKS ;ENABLE INTERRUPT
      NOP
      NOP
      CLR @TKS ;OK, CLEAN UP
      CHAIN ;EXIT TESTS
      BR 1$
      ERROR ;DLV INTERRUPTED WITH DONE CLEARED
      POPSP2 ;CLEAN UP THE STACK
      BR 10$ ;EXIT TESTS
  
```

CZLACFO LA36 TERM (DL11 & KL11)
CZLACF.P11 13-JUN-79 15:33

MACY11 30A(1052) 13-JUN-79^{E 6} 15:35 PAGE 69
I/O LOGIC TESTS

SEQ 0069

2897
2898

2899
 2900
 2901
 2902
 2903
 2904
 2905
 2906
 2907
 2908
 2909
 2910
 2911
 2912
 2913
 2914
 2915
 2916
 2917
 2918
 2919
 2920
 2921
 2922
 2923
 2924 007374 000000
 2925 007376 007450
 2926 007400 104016
 2927 007402 104007
 2928 007404 014627
 2929 007406 012703 025125
 2930 007412 012702 000004
 2931 007416 010300
 2932 007420 013701 000652
 2933 007424 104015
 2934 007426 000300
 2935 007430 005301
 2936 007432 001374
 2937 007434 000303
 2938 007436 104012
 2939 007440 005302
 2940 007442 001365
 2941 007444 104005
 2942 007446 000757

.SBITL LA36 PRINTER TESTS

: THE LA36 PRINTER TESTS WILL BE EXECUTED IN A
 : CONTINUOUS LOOP OUTPUTTING TO ALL MULTIPLE DL11'S
 : IF SR BIT 8 IS SET TO ZERO AT START UP TIME. IF
 : BIT 8 IS SET TO 1 AT START UP THEY MAY BE EXECUTED
 : INDIVIDUALLY ONCE OR CONTINUALLY LOOPED, OR
 : BECOME THE FIRST OF THE ENTIRE SEQUENCE OF PRINTER
 : TESTS. REFERENCE INTRUCTIONS IN THE INTRODUCTION
 : FOR PROPER MODE OF OPERATION.

:XXXXXXXXXX

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

```

:      *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
      TPEM       ;PRINT COLUMN # MMSG
      HDRO
1$:   MOV        #'U',R3    ;SET FIRST CHAR PAIR
      MOV        #4,R2     ;SET LINE COUNT
2$:   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 LINE START CHAR
      CRLF      ;SEND CR-LF
      DEC        R2        ;DEC LINE COUNT
      BNE        2$        ;FINISH TEST
      CHAIN     ;ALL DONE, EXIT
      BR        1$        ;REPEAT TEST
  
```

2943
2944
2945
2946
2947
2948
2949 007450 000001
2950 007452 007572
2951 007454 104016
2952 007456 012701 000040
2953 007462 012702 000100
2954 007466 012703 000140
2955 007472 110100
2956 007474 004737 007542
2957 007500 110200
2958 007502 004737 007542
2959 007506 012704 000003
2960 007512 110300
2961 007514 104015
2962 007516 005304
2963 007520 001375
2964 007522 104012
2965 007524 122122
2966 007526 105723
2967 007530 020327 000200
2968 007534 103756
2969 007536 104005
2970 007540 000746
2971 007542 012704 000003
2972 007546 104015
2973 007550 005304
2974 007552 001375
2975 007554 012700 000040
2976 007560 104015
2977 007562 012700 000040
2978 007566 104015
2979 007570 000207

```
;XXXXXXXXXX
:
: PT1 -- PRINTER CHARACTER TEST --- PRINTS ALL PRINTABLE CHARACTERS
:
:XXXXXXXXXX

PT1: 1 ;TEST NUMBER
      PT2 ;NEXT TEST
      PRTHDR

1$: MOV #40,R1 ;SPACE TO R1
     MOV #100,R2 ;@ TO R2
     MOV #140,R3 ;\ TO R3

2$: MOVE R1,R0 ;CHAR TO R0
     JSR PC,SPSP ;SEND TWO SPACES
     MOVB R2,R0 ;NEXT CHAR TO R0
     JSR PC,SPSP ;SEND TWO SPACES
     MOV #3,R4 ;PRINT COUNT TO R4
     MOVB R3,R0 ;THIRD CHAR TO R0

3$: PRINTC ;PRINT THE CHAR
     DEC R4 ;THREE TIMES ?
     BNE 3$ ;BRANCH IF NOT
     CRLF ;CARRIAGE RETURN LINE FEED
     CMPB (R1)+,(R2)+ ;NEXT CHARACTERS
     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 ;PRINT COUNT TO R4
1$: PRINTC ;PRINT CHAR
     DEC R4 ;THREE TIMES?
     BNE 1$ ;BRANCH IF NOT

SP2: MOV #40,R0 ;SPACE TO R0
     PRINTC ;SEND A SPACE

SPC: MOV #40,R0 ;SPACE TO R0
     PRINTC ;SEND ANOTHER
     RTS PC ;RETURN
```

2980				
2981				
2982				
2983				
2984				
2985				
2986				
2987				
2988				
2989				
2990				
2991				
2992				
2993	007572	000002		
2994	007574	010166		
2995	007576	104016		
2996	007600	012701	007700	
2997	007604	012703	010141	
2998	007610	012702	000003	
2999	007614	012704	000010	
3000	007620	121327	000055	
3001	007624	001422		
3002	007626	112100		
3003	007630	104015		
3004	007632	005304		
3005	007634	001371		
3006	007636	112300		
3007	007640	012704	000003	
3008	007644	104015		
3009	007646	005304		
3010	007650	001375		
3011	007652	005302		
3012	007654	001404		
3013	007656	004737	007554	
3014	007662	104015		
3015	007664	000753		
3016	007666	104012		
3017	007670	000747		
3018	007672	104012		
3019	007674	104005		
3020	007676	000740		
3021				
3022				
3023				
3024	007700	030060	020060	047040
3025	007706	046125	030060	020061
3026	007714	051440	044117	030060
3027	007722	020062	051440	054124
3028	007730	030060	020066	040440
3029	007736	045503	031060	020060
3030	007744	042040	042514	031060
3031	007752	020061	042040	030503
3032	007760	031060	020062	042040
3033	007766	031103	031060	020063
3034	007774	042040	031503	031060
3035	010002	020064	042040	032103

: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 TESTED.

:XXXXXXXXXX

```

PT2:      2          ;TEST NUMBER
          PT3       ;NEXT TEST
          PRTHDR    ;PRINT TEST HEADER
1$:      MOV      #IDEZ,R1 ;ADDR OF IDENT TO R1
          MOV      #NPCODE,R3 ;ADDR OF NON-PRINT-CODES TO R3
2$:      MOV      #3,R2   ;NO. OF ID'S PER LINE TO R2
3$:      MOV      #10,R4  ;NO. OF CHARS PER ID TO R4
4$:      CMPB    (R3),#55 ;ZERO TERMINATOR IN NP TABLE?
          BEQ     7$      ;BRANCH IF YES
          MOVB   (R1)+,R0 ;GET ID CHARACTERS
          PRINTC ;AND PRINT A
          DEC    R4       ;GROUP OF
          BNE   4$       ;8 CHARACTERS
          MOVB   (R3)+,R0 ;GET NP CODE FROM TABLE
          MOV    #3,R4   ;AND
5$:      PRINTC ;TRY TO PRINT IT
          DEC    R4       ;THREE
          BNE   5$       ;TIMES
          DEC    R2       ;MORE TO GO ON THIS LINE ?
          BEQ   6$       ;BRANCH IF NO
          JSR   PC,SP2   ;SEND 3 SPACES
          PRINTC
          BR    3$       ;BRANCH TO CONTINUE LINE
6$:      CRLF
          BR    2$       ;GO DO NEXT LINE
7$:      CRLF
          CHAIN
          BR    1$       ;CHAIN TO NEXT TEST

```

```

IDEZ:  .ASCII /000 NUL001 SOH002 STX/
       .ASCII /006 ACK020 DLE021 DC1/
       .ASCII /022 DC2023 DC3024 DC4/

```

3036	010010	031060	020065	047040					
3037	010016	045501	031060	020066	.ASCII	/025	NAK026	SYN027	ETB/
3038	010024	051440	047131	031060					
3039	010032	020067	042440	041124					
3040	010040	031460	020060	041440	.ASCII	/030	CAN031	EM 032	SUB/
3041	010046	047101	031460	020061					
3042	010054	042440	020115	031460					
3043	010062	020062	051440	041125					
3044	010070	031460	020064	043040	.ASCII	/034	FS 035	GS 036	RS /
3045	010076	020123	031460	020065					
3046	010104	043440	020123	031460					
3047	010112	020066	051040	020123					
3048	010120	031460	020067	052440	.ASCII	/037	US 177	DEL /	
3049	010126	020123	033461	020067					
3050	010134	042040	046105	040					
3051	010141	000	002	006	NPCODE:	.BYTE	0,2,6,20,21,22,23,24		
3052	010144	020	021	022					
3053	010147	023	024						
3054	010151	025	025	027	.BYTE	25,25,27,30,31,32,34,35			
3055	010154	030	031	032					
3056	010157	034	035						
3057	010161	036	037	177	.BYTE	36,37,177,55			
3058	010164	055							
3059	010166				.EVEN				

```

3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074 010166 000003
3075 010170 010306
3076 010172 104016
3077 010174 005037 000674
3078 010200 013701 000652
3079 010204 012700 000117
3080 010210 104015
3081 010212 005301
3082 010214 001404
3083 010216 004737 007562
3084 010222 005301
3085 010224 001367
3086 010226 104022
3087 010230 012737 000001 000674
3088 010236 013701 000674
3089 010242 004737 007562
3090 010246 005301
3091 010250 001374
3092 010252 012700 000130
3093 010256 104015
3094 010260 104022
3095 010262 062737 000002 000674
3096 010270 023737 000674 000652
3097 010276 103757
3098 010300 104014
3099 010302 104005
3100 010304 000733
  
```

```

:XXXXXXXXXX
:PT3 -- CARRIAGE RETURN TEST
  
```

```

: THE LINE CONSISTS OF A STRING OF O'S AND
: X'S. FIRST, THE O'S ARE PRINTED OUT TO THE LAST
: COLUMN WITH A SPACE SEPARATING EACH. THEN THE
: CARRIAGE IS SPACED TO THE FIRST BLANK SPACE, AN X
: IS PRINTED AND THEN RETURNED TO THE MARGIN. THIS
: PROCESS IS CONTINUE UNTIL ALL SPACES BETWEEN
: THE ZEROES HAVE BEEN FILLED.
  
```

```

:XXXXXXXXXX
  
```

```

PT3: 3 :TEST NUMBER
      PT4 :NEXT TEST
      PRTHDR :TYPE HEADER
1$: CLR SPCNT :CLEAR SPACE COUNTER
   MOV WIDTH,R1 :POSITION COUNTER TO R1
2$: MOV #117,R0 :"O" TO R0
   PRINTC :PRINT THE "O"
   DEC R1 :DECREMENT POSITION COUNTER
   BEQ 3$ :BRANCH IF 0
   JSR PC,SPC :SEND SPACE
   DEC R1 :DECREMENT POSITION COUNTER
   BNE 2$ :BRANCH IF NOT ZERO
3$: CR :SEND A CR
   MOV #1,SPCNT :SPACE, COUNTER SET TO 1
4$: MOV SPCNT,R1 :NO. OF SPACES TO R1
5$: JSR PC,SPC :SEND SPACE
   DEC R1 :DECREMENT SPACE COUNTER
   BNE 5$ :BRANCH IF NOT ZERO
   MOV #130,R0 :"X" INTO R0
   PRINTC :PRINT "X"
   CR :PRINT CR
   ADD #2,SPCNT :INCREMENT SPACE COUNT BY 2
   CMP SPCNT,WIDTH :COMPARE POSITION COUNTER WITH COLM. COUNT
   BLO 4$ :BRANCH IF LOWER
   LF :SEND LF
   CHAIN :CHAIN TO NEXT TEST
   BR 1$ :REPEAT TEST
  
```



```

3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111 010306 000004
3112 010310 010464
3113 010312 104016
3114 010314 012737 000001 000706
3115 010322 013701 000652
3116 010326 012702 010446
3117 010332 004737 010416
3118 010336 013701 000706
3119 010342 104014
3120 010344 005301
3121 010346 001375
3122 010350 006337 000706
3123 010354 022737 000100 000706
3124 010362 001406
3125 010364 112200
3126 010366 104015
3127 010370 112200
3128 010372 104015
3129 010374 104022
3130 010376 000757
3131 010400 013701 000652
3132 010404 004737 010416
3133 010410 104014
3134 010412 104005
3135 010414 000737
3136 010416 112200
3137 010420 104015
3138 010422 112200
3139 010424 104015
3140 010426 005741
3141 010430 012700 000137
3142 010434 104015
3143 010436 005301
3144 010440 001375
3145 010442 104022
3146 010444 000207
3147
3148 010446 030460 031060 032060
3149 010454 034060 033061 031063
3150 010462 030060

```

```

:XXXXXXXXXX
:PT4 -- MULTIPLE LINE FEED TEST -- 63 LINE FEEDS ARE
:SENT WITH A REFERENCE LINE AT THE START AND END.
:A NUMBER IS PRINTED WHICH INDICATES THE NUMBER OF LINE
:FEEDS THAT WILL BE ISSUED BEFORE THE NEXT
:NUMBER OR REFERENCE LINE IS PRINTED.
:XXXXXXXXXX
PT4: 4 :TEST NUMBER
PTS :NEXT TEST
PRTHDR :TYPE HEADER
1$: MOV #1,LFCNT :LINE FEED COUNT TO 1
MOV WIDTH,R1 :COLUMN COUNT TO R1
MOV #LINE3,R2 :ADDR OF NUMBER FIELD TO R2
JSR PC,REF :PRINT REFERENCE LINE
2$: MOV LFCNT,R1 :LINE FEED COUNT TO R1
3$: LF :SEND LF
DEC R1 :DECREMENT COUNTER
BNE 3$ :BRANCH IF NOT YET 0
ASL LFCNT :DOUBLE LINE FEED COUNT
CMP #BIT6,LFCNT :TEST IF COUNT IS 32
BEQ 4$ :BRANCH IF -32, END
MOVB (R2)+,RO :NUMBER TO RO
PRINTC :PRINT IT
MOVB (R2)+,RO :NUMBER TO RO
PRINTC :PRINT IT
CR :PRINT CR
BR 2$ :DRIVE THE LINEFEEDS
4$: MOV WIDTH,R1 :COLUMN COUNT TO R1
JSR PC,REF :SEND END REFERENCE LINE
LF :ADVANCE PAPER
CHAIN
BR 1$ :REPEAT TEST
REF: MOVB (R2)+,RO :NUMBER TO RO
PRINTC :PRINT IT
MOVB (R2)+,RO :NUMBER TO RO
PRINTC :PRINT IT
TST -(R1) :DECREASE COUNTER BY 2
MOV #137,RO :DASH (-) TO RO
1$: PRINTC :PRINT IT
DEC R1 :DECREMENT COLUMN COUNTER
BNE 1$ :BRANCH IF NO ZERO
CR :PRINT CR
RTS PC :RETURN

```

LINE 3: .ASCII /01020408163200/

3205
3206
3207
3208
3209
3210
3211
3212
3213
3214 010670 000006
3215 010672 011056
3216 010674 104016
3217 010676 104007
3218 010700 014627
3219 010702 013701 000652
3220 010706 005741
3221 010710 012700 000060
3222 010714 104015
3223 010716 005301
3224 010720 001375
3225 010722 012700 000062
3226 010726 104015
3227 010730 104015
3228 010732 023727 000652 000204
3229 010740 001404
3230 010742 012700 003410
3231 010746 104010
3232 010750 000407
3233 010752 012700 000063
3234 010756 012701 000100
3235 010762 104015
3236 010764 005301
3237 010766 001375
3238 010770 104012
3239 010772 013701 000652
3240 010776 012700 000134
3241 011002 104015
3242 011004 012700 000010
3243 011010 104015
3244 011012 012700 000057
3245 011016 104015
3246 011020 005301
3247 011022 001365
3248 011024 104014
3249 011026 104022
3250 011030 004737 010632
3251 011034 104012
3252 011036 012700 001750
3253 011042 104010
3254 011044 004737 010632
3255 011050 104012
3256 011052 104005
3257 011054 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 HEADER
      TYPEN     ;PRINT COLUMN # MSG
      HDRO
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$
      MOV      #3410,R0 ;DELAY 1.8 SEC
      DELAY
      BR       5$
3$:   MOV      #63,R0   ;3'S TO R0
      MOV      #100,R1  ;64 TO CHAR COUNT
4$:   PRINTC   ;SEND CHAR
      DEC      R1       ;DECREMENT CHAR COUNT
      BNE      4$      ;CONTINUE IF NOT DONE
      CRLF     ;SEND A CR,LF
5$:   MOV      WIDTH,R1 ;COLUMN COUNT TO R1
      MOV      #134,R0  ;BACKSLASH TO R0
      PRINTC   ;SEND IT
      MOV      #10,R0   ;BACKSPACE TO R0
      PRINTC   ;SEND IT
      MOV      #57,R0   ;FORWARD SLASH TO R0
      PRINTC   ;SEND IT
      DEC      R1       ;END OF PAPER
      BNE      6$      ;BRANCH IF NO
      LF
      CR
      JSR      PC,PT5AL ;SEND REF LINE #1
      CRLF     ;SEND A CR,LF
6$:   MOV      #1750,R0 ;DELAY 1 SEC
      DELAY
      JSR      PC,PT5AL ;SEND SECOND REF LINE
      CRLF     ;SEND A CR,LF
      CHAIN   ;CHAIN TO NEXT TEST
      BR       1$     ;REPEAT TEST

```

065

3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269 011056 000007
3270 011060 011270
3271 011062 104016
3272 011064 012703 000002
3273 011070 013701 000652
3274 011074 012700 000115
3275 011100 104015
3276 011102 005301
3277 011104 001404
3278 011106 004737 007562
3279 011112 005301
3280 011114 001367
3281 011116 022703 000002
3282 011122 001003
3283 011124 104022
3284 011126 005303
3285 011130 000757
3286 011132 005703
3287 011134 001373
3288 011136 104012
3289 011140 005723
3290 011142 013701 000652
3291 011146 004737 007562
3292 011152 005301
3293 011154 001405
3294 011156 012700 000100
3295 011162 104015
3296 011164 005301
3297 011166 001367
3298 011170 022703 000002
3299 011174 001003
3300 011176 104022
3301 011200 005303
3302 011202 000757
3303 011204 005703
3304 011206 001373
3305 011210 104012
3306 011212 005723
3307 011214 013701 000652
3308 011220 012700 000046
3309 011224 104015
3310 011226 005301
3311
3312 011230 001404
3313 011232 004737 007562

```

:XXXXXXXXXX
:PT7-- OVERPRINT TEST-- A ROW OF ALTERNATING M'S AND
:      SPACES ARE PRINTED, OUT TO THE LAST COLUMN AND OVERPRINTED TWICE.
:      A SECOND LINE OF ALTERNATING SPACES AND "Q'S" IS THEN
:      SENT 3 TIMES AS THE FIRST LINE. THIS IS FOLLOWED
:      BY A THIRD AND FINAL LINE OF ALTERNATING 'B'
:      AND SPACES.
:XXXXXXXXXX

```

```

PT7: 7 ;TEST NUMBER
      PT10 ;NEXT TEST
      PRTHDR ;PRINT MESSAGE
1$: MOV #2,R3 ;2 COUNT TO R3
2$: MOV WIDTH,R1 ;NO. OF COLUMNS TO R1
3$: MOV #115,R0 ;'M' TO R0
      PRINTC ;SEND IT
      DEC R1 ;END OF LINE
      BEQ 4$ ;BRANCH IF YES
      JSR PC,SPC ;SEND SPACE
      DEC R1 ;END OF LINE?
      BNE 3$ ;BRANCH IF NO
4$: CMP #2,R3 ;TEST R3
      BNE 6$ ;BRANCH IF NOT FIRST TIME
5$: CR ;SEND CR
      DEC R3 ;DECREASE LINE COUNTER
      BR 2$ ;REPEAT LINE
6$: TST R3 ;THIRD TIME?
      BNE 5$ ;BRANCH IF NOT
      CRLF ;NEXT LINE
      TST (R3)+ ;REPEAT COUNTER TO R3
7$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
8$: JSR PC,SPC ;SEND SPACE
      DEC R1 ;DECREASE COLUMN COUNT
      BEQ 9$ ;BRANCH IF 0, END OF LINE
      MOV #100,R0 ;'Q' TO R0
      PRINTC ;SEND IT
      DEC R1 ;DECREASE COLUMN COUNT
      BNE 8$ ;BRANCH IF NOT 0 (NOT END)
9$: CMP #2,R3 ;END OF LINE, FIRST TIME?
      BNE 11$ ;BRANCH IF NOT
10$: CR ;SEND CR
      DEC R3 ;DECREASE LINE COUNTER
      BR 7$ ;REPEAT LINE
11$: TST R3 ;TEST IF THIRD REPEAT
      BNE 10$ ;BRANCH IF NOT
      CRLF ;DO NEXT LINE
      TST (R3)+ ;LINE REPEAT COUNTER TO R3
12$: MOV WIDTH,R1 ;COLUMN COUNT TO R1
13$: MOV #46,R0 ;'B' TO R0
      PRINTC ;SEND IT
      DEC R1 ;DECREASE COLUMN COUNT
      BEQ 14$ ;BRANCH IF END
      JSR PC,SPC ;SEND SPACE

```

3314 011236 005301
3315 011240 001367
3316 011242 022703 000002
3317 011246 001003
3318 011250 104022
3319 011252 005303
3320 011254 000757
3321 011256 005703
3322 011260 001373
3323 011262 104012
3324 011264 104005
3325 011266 000676

14\$: DEC R1 :DECREASE COLUMN COUNT
BNE 13\$:BRANCH IF NOT END
CMP #2,R3 :TEST IF FIRST TIME
BNE 16\$:BRANCH IF =2, FIRST TIME
15\$: CR :SENT CR
DEC R3 :DECREASE REPEAT COUNTER
BR 12\$:PRINT LINE AGAIN
16\$: TST R3 :TEST IF END, R3=0
BNE 15\$:BRANCH IF NOT END
CRLF :SEND CR,LF
CHAIN :CHAIN TO NEXT TEST
BR 1\$:REPEAT TEST

3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339 011270 000010
3340 011272 011426
3341 011274 104016
3342 011276 012701 000036
3343 011302 012702 000170
3344 011306 012737 000036 011324
3345 011314 012700 000110
3346 011320 104015
3347 011322 012700 000036
3348 011326 104010
3349 011330 005301
3350 011332 001426
3351 011334 005302
3352 011336 001430
3353 011340 013704 011324
3354 011344 006204
3355 011346 006204
3356 011350 006204
3357 011352 006204
3358 011354 010405
3359 011356 006204
3360 011360 006204
3361 011362 006204
3362 011364 060405
3363 011366 022702 000074
3364 011372 003403
3365 011374 160537 011324
3366 011400 000745
3367 011402 060537 011324
3368 011406 000742
3369 011410 104012
3370 011412 012701 000036
3371 011416 000746
3372 011420 104012
3373 011422 104005
3374 011424 000724

:XXXXXXXXXX

10-- 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;

$$\text{NEW DELAY} = \text{OLD DELAY} [+ \text{OR} -] (\text{OLD DELAY}/16 + \text{OLD DELAY}/128)$$

:XXXXXXXXXX

```

PT10: 10          ;TEST NUMBER
        PT11       ;NEXT TEST
        PRTHDR     ;TYPE MESSAGE
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        R1         ;DEC. COUNT OF CHARS PER 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

```

3375
3376
3377
3378
3379
3380
3381
3382
3383
3384 011426 000011
3385 011430 011460
3386 011432 104016
3387 011434 012701 000030
3388 011440 012700 000130
3389 011444 104015
3390 011446 104012
3391 011450 005301
3392 011452 001372
3393 011454 104005
3394 011456 000766
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407 011460 000012
3408 011462 007374
3409 011464 104016
3410 011466 012701 000010
3411 011472 012700 000007
3412 011476 104015
3413 011500 005301
3414 011502 001375
3415 011504 104014
3416 011506 012700 003720
3417 011512 104010
3418 011514 013700 000042
3419 011520 001405
3420 011522 000240
3421 011524 004710
3422 011526 000240
3423 011530 000240
3424 011532 000240
3425 011534 104005
3426 011536 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 FEED MECHANISM FOR PROPER OPERATION.
:XXXXXXXXXX

PT11: 11 ;TEST NUMBER
PT12 ;NEXT TEST
PRTHDR ;TYPE MESSAGE
1\$: MOV #30,R1 ;SET R1=24(10), LINE COUNT
2\$: MOV #130,R0 ;SET CHAR = X
PRINTC ;PRINT X
CRLF ;SEND CR-LF
DEC R1 ;DECREMENT LINE COUNT
BNE 2\$;CONTINUE 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 ;TYPE HEADER
PT12A: MOV #10,R1 ;COUNTER TO R1
MOV #7,R0 ;BELL TO R0
1\$: PRINTC ;SEND IT
DEC R1 ;DECREMENT COUNT
BNE 1\$;BRANCH IF NOT ZERO
LF ;
MOV #3720,R0 ;DELAY 2 SEC BEFORE RESTARTING
DELAY ;
MOV @#42,R0 ;CHECK IF UNDER ACT11 OR XXDP
BEQ HERE ;CONTINUE TEST SEQUENCE
NOP ;A RESET WAS FORMERLY HERE
LOGICAL: JSR PC,(R0)
NOP
NOP
NOP
HERE: CHAIN ;CHAIN TO NEXT TEST
BR PT12A ;REPEAT TEST

3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438 011540 000017
3439 011542 011540
3440 011544 000137 011600
3441 011550 000017
3442 011552 011540
3443 011554 005037 012116
3444 011560 013704 000652
3445 011564 012737 000001 012114
3446 011572 104016
3447 011574 104007
3448 011576 014627
3449 011600 012703 000041
3450 011604 005237 012116
3451 011610 023727 012116 000031
3452 011616 001003
3453 011620 012737 000001 012116
3454 011626 012700 014542
3455 011632 013701 012116
3456 011636 012702 000002
3457 011642 104023
3458 011644 013701 000652
3459 011650 010300
3460 011652 004737 011766
3461 011656 104015
3462 011660 005301
3463 011662 003372
3464 011664 004737 012034
3465 011670 104012
3466 011672 012702 000005
3467 011676 013701 000652
3468 011702 010300
3469 011704 004737 011766
3470 011710 104015
3471 011712 005301
3472 011714 003372
3473 011716 104022
3474 011720 005302
3475 011722 001365
3476 011724 004737 012034
3477 011730 104014
3478 011732 005203
3479 011734 022703 000177
3480 011740 001341
3481 011742 004737 012034
3482 011746 004737 012034

```
;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 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
MOV WIDTH,R4 ;INITIALIZE R4
MOV #1,DIRTN ;AND DIRECTION OF PRECESS
PRTHDR ;PRINT COLUMN # MMSG
TYPEM
HDRO
PT17D: MOV #41,R3 ;SET START CHAR
INC PASCNT
CMP PASCNT,#31 ;DO 31 TIMES
BNE 20$ ;BRANCH IF NOT DONE
MOV #1,PASCNT ;START OVER
20$: MOV #PASMES,R0 ;SET MMSG ADDR
MOV PASCNT,R1 ;# TO CONVERT
MOV #2,R2 ;# DIGITS
BTOASC ;CONVERT PASCNT TO ASCII
1$: MOV WIDTH,R1 ;SET COLUMN COUNT
2$: MOV R3,R0 ;GET CHARACTER
JSR PC,CKPOS ;TIME TO INSERT PASS # ?
PRINTC ;SEND CHAR
DEC R1 ;DECREMENT COUNT
BGT 2$ ;BRANCH IF NOT DONE
JSR PC,ADJR4 ;ADJUST R4 POINTER
CRLF
MOV #5,R2 ;SET OVERPRINT COUNT
3$: MOV WIDTH,R1 ;SET COLUMN COUNT
4$: MOV R3,R0 ;GET CHARACTER
JSR PC,CKPOS ;TIME TO INSERT PASS # ?
PRINTC ;SEND CHAR
DEC R1 ;DECREMENT COUNT
BGT 4$ ;BRANCH IF NOT DONE
CR ;SEND CR
DEC R2 ;DONE OVERPRINTS ?
BNE 3$ ;NO. CONTINUE
JSR PC,ADJR4 ;ADJUST R4 POINTER
LF ;SEND LF
INC R3 ;SET NEXT CHAR
CMP #177,R3 ;DONE CHAR SET ?
BNE 1$ ;NO. CONTINUE
JSR PC,ADJR4 ;OFFSET POINTER 3 PLACES
JSR PC,ADJR4 ;TO RETAIN VISUAL ALIGNMENT
```


CZLACFO LA36 TERM (DL11 & KL11)
CZLACF.P11 13-JUN-79 15:33

MACY11 30A(1052) 13-JUN-79^{F 7} 15:35 PAGE 83
LA36 PRINTER TESTS

SEQ 0083

3483 011752 004737 012034
3484 011756 104007
3485 011760 014523
3486 011762 104005
3487 011764 000705
3488

JSR PC,ADJR4
TYPEM
ENDPAS
CHAIN
BR PT17D

:THROUGH END OF PASS
:TYPE END OF PASS MESSG
:REPEAT TEST

3489
3490 011766 020401
3491 011770 001020
3492 011772 012700 000040
3493 011776 104015
3494 012000 113700 014542
3495 012004 104015
3496 012006 113700 014543
3497 012012 104015
3498 012014 012700 000040
3499 012020 104015
3500 012022 162701 000003
3501 012026 062716 000002
3502 012032 000207
3503
3504 012034 005737 012114
3505 012040 001013
3506 012042 005204
3507 012044 020437 000652
3508 012050 101420
3509 012052 013704 000652
3510 012056 005304
3511 012060 012737 000001 012114
3512 012066 000411
3513 012070 005304
3514 012072 020427 000004
3515 012076 002401
3516 012100 000404
3517 012102 012704 000005
3518 012106 005037 012114
3519 012112 000207
3520
3521 012114 000000
3522
3523 012116 000000

```
CKPOS:  CMP      R4,R1      ;IS IT TIME TO INSERT PASS # ?
        BNE      1$          ;BRANCH IF NO
        MOV      #40,R0      ;PRINT A SPACE
        PRINTC
        MOV      PASMES,R0   ;PRINT MSG OF PASS COUNT
        PRINTC
        MOV      PASMES+1,R0
        PRINTC
        MOV      #40,R0      ;PRINT A SPACE
        PRINTC
        SUB      #3,R1        ;ADJUST R1 3 POSITIONS
        ADD      #2,(SP)      ;ADJUST RETURN PC OVER PRINTC
1$:      RTS      PC

ADJR4:  TST      DIRTN       ;TEST DIRECTION OF PRECESS
        BNE      1$          ;BR IF LEFT
        INC      R4           ;INCREASE POSITION CNTR
        CMP      R4,WIDTH    ;IS R4 > WIDTH ?
        BLOS    3$           ;BR IF NOT GREATER
        MOV      WIDTH,R4    ;CHANGE DIRECTION
        DEC      R4           ;      TO
        MOV      #1,DIRTN    ;      LEFT.
        BR      3$

1$:      DEC      R4           ;DECREASE POSITION CNTR
        CMP      R4,#4       ;LESS THAN 4 ?
        BLT     2$           ;BR IF YES
        BR      3$           ;ELSE EXIT
2$:      MOV      #5,R4      ;SET R4 TO POS 5
        CLR      DIRTN      ;CHANGE DIRECTION TO RIGHT
3$:      RTS      PC        ;EXIT

DIRTN:  .WORD   0           ;DIRECTION OF PRECESS (0 LEFT)

PASCNT: .WORD   0
```

CZLACFO LA36 TERM (DL11 & KL11)
CZLACF.P11 13-JUN-79 15:33

MACY11 30A(1052) 13-JUN-79^{M 7} 15:35 PAGE 85
LA36 PRINTER TESTS

SEQ 0085

3524

3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537 012120 000020
3538 012122 012170
3539 012124 104016
3540 012126 104020
3541 012130 012700 0C0036
3542 012134 104010
3543 012136 022737 000177 000700
3544 012144 001405
3545 012146 104017
3546 012150 117777 166440 166442
3547 012156 000763
3548 012160 104007
3549 012162 014772
3550 012164 104005
3551 012166 000757
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563 012170 000021
3564 012172 012226
3565 012174 104016
3566 012176 012737 000060 000662
3567 012204 013702 000652
3568 012210 013700 000662
3569 012214 104015
3570 012216 005302
3571 012220 003373
3572 012222 104012
3573 012224 000767

.SBTTL LA36 ECHO TESTS

: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
E021 ;NEXT TEST
PRTHDR ;TYPE HEADER
1\$: READ ;GO WAIT FOR KEYBOARD INPUT
MOV #30.,R0 ;DELAY FOR HALF DUPLEX
DELAY
CMP #177,TEMPCH ;CHECK IF RUBOUT
BEQ 2\$;BRANCH IF YES
PRNT ;NO, CHECK PRINTER READY
MOVB @TKB,@TPB ;READY, ECHO CHARACTER
BR 1\$
2\$: TYPEM ;PRINT TERMINATION MESSAGE
E0END
CHAIN ;CHAIN TO NEXT TEST
BR 1\$;REPEAT TEST

:XXXXXXXXXX

:E021-- LINE ECHO TEST, FAST RATE-- THIS TEST WILL
CAUSE THE CONTINUAL PRINTING OF 'O' AT THE MAXIMUM
RATE UNTIL EITHER ANOTHER CHARACTER IS SELECTED
BY PRESSING A KEY ON THE KEYBOARD OR TERMINATION BY THE
RUBOUT.

:XXXXXXXXXX

E021: 21 ;TEST NUMBER
E022 ;NEXT TEST
PRTHDR ;TYPE HEADER
E021A: MOV #60,REPT ;CHARACTER TO BE REPEATED (O)
1\$: MOV WIDTH,R2 ;SET COLUMN COUNT
2\$: MOV REPT,R0 ;GET CHAR
PRINTC ;PRINT CHAR
DEC R2 ;DEC COLUMN COUNT
BGT 2\$;FINISH LINE
CRLF ;SEND A CR AND LF
BR 1\$

3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595

012226 000022
012230 012500
012232 104016
012234 012737 000060 000662
012242 013702 000652
012246 013700 000662
012252 104015
012254 005302
012256 001404
012260 012700 003410
012264 104010
012266 000767
012270 104012
012272 000763

```
:XXXXXXXXXX
:
:EO22-- LINE ECHO TEST, SLOW RATE-- SAME AS EO21 EXCEPT
:          THAT A DELAY IS INTRODUCED BETWEEN CHARACTERS
:          TO PRODUCE A LCV ACTION
:
:XXXXXXXXXX
:EO22: 22
:      EQ23
:      PRTHDR
:EO22A: MOV #60,REPT      ;TYPE HEADER
:1$:    MOV WIDTH,R2     ;LOAD 0 AS INITIAL CHARACTER
:2$:    MOV REPT,R0      ;SET COLUMN COUNT
:      PRINTC           ;GET CHAR
:      DEC R2           ;PRINT CHAR
:      BEQ 3$           ;DEC COLUMN COUNT
:      MOV #3410,R0     ;BRANCH IF DONE LINE
:      DELAY           ;DELAY 1.8 SEC.
:      BR 2$           ;OUTPUT NEW CHAR.
:3$:    CRLF            ;SEND A CR AND LF
:      BR 1$
```

3596
3597
3598
3599
3600
3601
3602 012274 052516 020114
3603 012300 047523 020110
3604 012304 052123 020130
3605 012310 052105 020130
3606 012314 047505 020124
3607 012320 047105 020121
3608 012324 041501 020113
3609 012330 042502 020114
3610 012334 051502 020040
3611 012340 052110 020040
3612 012344 043114 020040
3613 012350 052126 020040
3614 012354 043106 020040
3615 012360 051103 020040
3616 012364 047523 020040
3617 012370 044523 020040
3618 012374 046104 020105
3619 012400 041504 020061
3620 012404 041504 020062
3621 012410 041504 020063
3622 012414 041504 020064
3623 012420 040516 020113
3624 012424 054523 020116
3625 012430 052105 020102
3626 012434 040503 020116
3627 012440 046505 020040
3628 012444 052523 020102
3629 012450 051505 020103
3630 012454 051506 020040
3631 012460 051507 020040
3632 012464 051522 020040
3633 012470 051525 020040
3634 012474 050123 020040
3635
3636

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

MONIC: .ASCII /NUL /
.ASCII /SOH /
.ASCII /STX /
.ASCII /ETX /
.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

```

3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648 012500 000023
3649 012502 013022
3650 012504 104016
3651 012506 104020
3652 012510 012700 000036
3653 012514 104010
3654 012516 023727 000700 000041
3655 012524 103015
3656 012526 004737 012662
3657 012532 113700 000700
3658 012536 006300
3659 012540 006300
3660 012542 062700 012274
3661 012546 004737 012740
3662 012552 104000
3663 012554 015021
3664 012556 000753
3665 012560 023727 000700 000177
3666 012566 001421
3667 012570 012701 013012
3668 012574 113721 000700
3669 012600 112721 000040
3670 012604 112721 000040
3671 012610 112721 000040
3672 012614 004737 012662
3673 012620 012700 013012
3674 012624 004737 012740
3675 012630 000750
3676 012632 004737 012662
3677 012636 012700 013016
3678 012642 004737 012740
3679 012646 104000
3680 012650 015021
3681 012652 104007
3682 012654 014772
3683 012656 104005
3684 012660 000712
3685 012662 012702 000003
3686 012666 012701 015023
3687 012672 062701 000003
3688 012676 013700 000704
3689 012702 042700 177770
3690 012706 062700 000060
3691 012712 110041
3692 012714 005302

```

```

:XXXXXXXXXX
:E023-- CHARACTER CODE TEST-- ANY CHARACTER SELECTED
:          WILL BE ECHCED ALONG WITH ITS OCTAL CODE.
:          A MNEMONIC WILL BE PRINTED INSTEAD OF THE CHARACTER
:          IF IT IS A NON-PRINTING CHARACTER.
:          THE PARITY OF THE RECEIVED CODE WILL ALSO BE
:          INDICATED AS EITHER EVEN OR ODD.
:XXXXXXXXXX
E023: 23          ;TEST NUMBER
      E024          ;NEXT TEST
      PRTHDR        ;TYPE HEADER
1$:  READ          ;GO WAIT FOR CHARACTER
      MOV #30.,RO   ;DELAY FOR HALF DUPLEX
      DELAY
      CMP TEMPCH,#41 ;TEST IF CHAR IS PRINTABLE
      BHIS 3$       ;BRANCH IF IT IS
      JSR PC,STRLN  ;STORE CODE INTO MESSAGE
      MOVB TEMPCH,RO ;GET CODE AGAIN
      ASL RO        ;MULT BY 2
      ASL RO        ;MULT BY 4
      ADD #MONIC,RO ;ADD ADDR OF MNEMONIC TABLE
      JSR PC,MOVNUM ;MOV MNEMONIC TO MESSAGE
2$:  TYPE          ;TYPE CODE AND MNEMONIC
      E023M        ;ADDRESS OF MESSAGE
      BR 1$         ;GO WAIT FOR NEXT CHARACTER
3$:  CMP TEMPCH,#177 ;TEST IF CHAR IS A RUBOUT
      BEQ 4$        ;BRANCH IF RUBOUT
      MOV #MG24,R1
      MOVB TEMPCH,(R1)+
      MOVB #40,(R1)+
      MOVB #40,(R1)+
      MOVB #40,(R1)+
      JSR PC,STRLN  ;STORE CODE INTO MESSAGE
      MOV #MG24,RO  ;ADDR OF CHAR INTO RO
      JSR PC,MOVNUM ;MOVE CHAR INTO MESSAGE
      BR 2$         ;TYPE MESSAGE
4$:  JSR PC,STRLN  ;RUBOUT, CONVERT AND STOR CODE
      MOV #MG25,RO  ;ADDR. OF DEL INTO RO
      JSR PC,MOVNUM ;MOVE DEL INTO MESSAGE
      TYPE          ;TYPE MESSAGE
      E023M        ;ADDR OF MESSAGE
      E023M        ;ADDR OF MESSAGE
      CHAIN        ;CHAIN TO NEXT TEST
      BR 1$         ;REPEAT TEST
STRLN: MOV #3,R2    ;COUNT OF 3 TO R2
      MOV #LINE5,R1 ;ADDR OF MESSG TO R1
      ADD #3,R1     ;POINT TO LAST SPACE IN MESSG
1$:  MOV PCHAR,RO   ;MOVE OCTAL CODE TO RO
      BIC #177770,RO ;SAVE LS OCTAL CHAR
      ADD #60,RO    ;MAKE ASCII
      MOVB RO,-(R1) ;MOVE INTO MESSG
      DEC R2        ;DECREMENT CHAR COUNTER

```



```

3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735 013022 000024
3736 013024 013570
3737 013026 104016
3738 013030 005001
3739 013032 012702 013166
3740 013036 104020
3741 013040 012700 000036
3742 013044 104010
3743 013046 022737 000177 000700
3744 013054 001440
3745 013056 022737 000003 000700
3746 013064 001413
3747 013066 020127 000400
3748 013072 103361
3749 013074 113722 000700
3750 013100 005201
3751 013102 104017
3752 013104 113777 000700 165506
3753 013112 000751
3754
3755
3756
3757 013114 020227 013166
3758 013120 001403
3759 013122 113722 000700
3760 013126 104013
3761 013130 012702 013166
3762 013134 021227 000003
3763 013140 001733
3764 013142 112200
3765 013144 020027 000003
3766 013150 001767
3767 013152 104015
3768 013154 000772
3769 013156 104007
3770 013160 014772
3771 013162 104005
3772 013164 000721
3773 013166 000003
3774 013170 000400

```

```

:XXXXXXXXXX
:E024-- SELECTED 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 CONTINOUS STRING UNTIL EITHER THE RUBOUT IS
: SELECTED TO TERMINATE OR THE CNTL/C IS SELECTED
: AGAIN. A TERMINATING CNTL/C FOLLOWED BY ANOTHER
: CNTL/C WILL ALWAYS CAUSE THE LAST INPUTTED STRING TO
: BE PRINTED. A TERMINATING CNTL/C FOLLOWED BY A CHARACTER OTHER THAN A
: RUBOUT WILL CAUSE A NEW STRING TO BE INPUTTED.
:XXXXXXXXXX

```

```

E024: 24 ;TEST NUMBER
      E025 ;NEXT TEST
      PR.HDR ;TYPE TEST HEADER
E024B: CLR R1 ;CLEAR CHARACTER COUNT
      MOV #BUFR,R2 ;ADDRESS OF BUFFER TO R2
1$: READ ;WAIT FOR INPUT
      MOV #30.,R0 ;DELAY FOR HALF DUPLEX
      DELAY
      CMP #177,TEMPCH ;TEST IF RUBOUT
      BEQ TERM ;BRANCH IF RUBOUT
      CMP #3,TEMPCH ;TEST IF CNTL-C
      BEQ OUTPUT ;BRANCH IF CNTL-C
      CMP R1,#256. ;YES, CHECK IF CHAR CNT IS EQ, GT 256
      BHIS 1$ ;BRANCH IF YES, IGNORE CHAR
      MOVB TEMPCH,(R2)+ ;STORE CHAR INTO BUFFER
      INC R1 ;INCREMENT CHARACTER COUNT
      PRNT ;CHECK IF PRINTER READY
      MOVB TEMPCH,@TPB ;ECHO CHAR
      BR 1$ ;GO WAIT FOR NEXT CHAR

;SECTION TO OUTPUT CONTINOUS STRING
OUTPUT: CMP R2,#BUFR ;CHECK IF POINTER IS AT START OF TABLE
        BEQ 1$ ;YES, BRANCH
        MOVB TEMPCH,(R2)+ ;NO, STORE ^C IN TABLE
        SCRLF ;SEND A CR LF
1$: MOV #BUFR,R2 ;BUFFER ADDRESS TO R2
      CMP (R2),#3 ;CHECK IF FIRST CHAR IS ^C
      BEQ E024B ;YES, LOOK FOR INPUT AGAIN
2$: MOVB (R2)+,R0 ;GET CHARACTER
      CMP R0,#3 ;DONE STRING?
      BEQ 1$ ;YES, RESTART STRING
      PRINTC ;PRINT CHAR
      BR 2$ ;CONTINUE
TERM: TYPEM ;OUTPUT TERMINATION MESSAGE
      ECOEND
      CHAIN ;CHAIN TO NEXT TEST
      BR E024B ;REPEAT TEST
BUFR: 3 ;INITIALIZE FIRST CHAR AS CNTL-C IN TABLE
      .BLKB 256. ;256 CHARACTER BUFFER

```

3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786 013570 000025
3787 013572 012120
3788 013574 104016
3789 013576 023727 000652 000101
3790 013604 103427
3791 013606 104007
3792 013610 014645
3793 013612 000402
3794 013614 104000
3795 013616 014645
3796 013620 104020
3797 013622 012700 000036
3798 013626 104010
3799 013630 023727 000700 000040
3800 013636 103770
3801 013640 022737 000177 000700
3802 013646 001410
3803 013650 104017
3804 013652 113777 000700 164740
3805 013660 104013
3806 013662 000754
3807 013664 104007
3808 013666 014745
3809 013670 104007
3810 013672 014772
3811 013674 104005
3812 013676 000737

:XXXXXXXXXXXX

: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.

:XXXXXXXXXXXX

```

E025: 25 ;TEST NUMBER
      E020 ;NEXT TEST HEADER
      PRTHDR ;PRINT HEADER
      CMP WIDTH,#101 ;TEST IF COLUMN COUNT IS EQ,GT 64
      BLO 4$ ;BRANCH IF NOT
      TYPEM ;TYPE TEST MESG
      E025MA
      BR 3$ ;WAIT FOR CHAR
      TYPE ;TYPE TEST MESG ON TERM CHAR RCVD ON
      E025MA
      READ ;WAIT FOR OPERATOR RESPONSE
      MOV #30.,R0 ;DELAY FOR HALF DUPLEX
      DELAY
      CMP TEMPCH,#40 ;TEST IF PRINTABLE
      BLO 3$ ;BRANCH IF NON-PRINTABLE
      CMP #177,TEMPCH ;CHECK IF CHAR IS RUBOUT
      BEQ 5$ ;BRANCH IF YES
      PRNT ;CHECK IF PRINTER IS READY
      MOVB TEMPCH,@TPB ;PRINT CHAR. (BELL SHOULD SOUND)
      SCRLF ;SEND A CRLF
      BR 2$ ;REPEAT
      TYPEM ;TYPE ERROR MESSAGE
      E025MB
      TYPEM ;PRINT TERMINATION
      ECOEND
      CHAIN ;EXIT TO NEXT TEST
      BR 1$ ;REPEAT TEST

```

```
3813  
3814  
3815  
3816  
3817  
3818  
3819  
3820  
3821 013700  
3822 013700 022737 000176 000714  
3823 013706 001132  
3824 013710 105777 164676  
3825 013714 100127  
3826 013716 017737 164672 000722  
3827 013724 042737 177600 000722  
3828 013732 022737 000007 000722  
3829 013740 001115  
3830 013742 104000  
3831 013744 015200  
3832 013746 005077 164646  
3833 013752 104000  
3834 013754 015206  
3835 013756 017746 164732  
3836 013762 004737 014214  
3837 013766 022600  
3838 013770 104000  
3839 013772 015216  
3840 013774 005037 000724  
3841 014000 012737 000007 000726  
3842 014006 105777 164600  
3843 014012 100375  
3844 014014 117737 164574 000722  
3845 014022 105777 164570  
3846 014026 100375  
3847 014030 113777 000722 164562  
3848 014036 042737 177600 000722  
3849 014044 122737 000025 000722  
3850 014052 001001  
3851 014054 000736  
3852 014056 122737 000015 000722  
3853 014064 001005  
3854 014066 104013  
3855 014070 022737 000007 000726  
3856 014076 001033  
3857 014100 122737 000060 000722  
3858 014106 003004  
3859 014110 122737 000067 000722  
3860 014116 002003  
3861 014120 104000  
3862 014122 015226  
3863 014124 000753  
3864 014126 006337 000724  
3865 014132 006337 000724  
3866 014136 006337 000724  
3867 014142 142737 000060 000722  
3868 014150 153737 000722 000724
```

```
:XXXXXXXXXXXXXXXXX  
:CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW  
:CHANGING OF LOC 176.  
:XXXXXXXXXXXXXXXXX  
:*****  
CHGFS:  
CKSWR:  CMP #SWREG,SR ;SOFTWARE SW REG PRES?  
        BNE OUT ;NO,GET OUT  
        TSTB @TKS ;YES, IS CHARACTER READY?  
        BPL OUT ;IF NOT, GET OUT  
        MOV @TKB,TIB ;STORE BUFFER  
        BIC #177600,TIB ;STRIP OFF GARBAGE  
        CMP #7,TIB ;IS IT A ^G  
        BNE OUT ;IF NOT GET OUT  
        TYPE ;ECHO ^G  
CNTLU:  CLR @TPB ;ALLOW SWR- TO BE TYPED  
        SWR  
        MOV @SR,-(SP) ;MOV CONTENTS OF SWR  
        JSR PC,TYPEOUT ;OCTAL TYPE OUT ROUTINE  
        CMP (%)+,R0 ;CORRECT STACK POINTER  
        TYPE ;ALLOW NEW= TO BE TYPED  
        NEW  
        CLR TEMPST ;CLEAR TEMP STORAGE LOC  
        MOV #7,COUNT ;SET UP TO ACCEPT 7 CHAR  
1$:     TSTB @TKS ;IS CHARACTER THERE?  
        BPL 1$ ;IF NOT, TRY AGAIN  
        MOV @TKB,TIB ;PICKUP CHARACTER  
8$:     TSTB @TPS ;CHECK PRINTER STATUS  
        BPL 8$ ;NOT READY, TRY AGAIN  
        MOV TIB,@TPB ;PRINT IT  
        BIC #177600,TIB ;STRIP OFF GARBAGE  
        CMPB #25,TIB ;IS IT A ^U  
        BNE 2$ ;BRANCH IF NOT  
3$:     BR CNTLU ;START OVER  
2$:     CMPB #15,TIB ;IS IT A <CR>  
        BNE 4$ ;BRANCH IF NOT  
        SCRLF  
        CMP #7,COUNT ;WAS <CR> FIRST CHAR  
        BNE 7$ ;CHANGE SWREG IF NOT FIRST <CR>  
4$:     CMPB #60,TIB ;IS IT LESS THAN 0  
        BGT 5$ ;GO TO ? ROUTINE IF SO  
        CMPB #67,TIB ;IS IT GREATER THAN 7  
        BGE 6$ ;GO TO ? ROUTINE IF SO  
5$:     TYPE ;SET UP FOR ? TYPEOUT  
        QUEST  
        BR 3$ ;START INPUT STRING OVER  
6$:     ASL TEMPST ;MULTIPLY BY 10  
        ASL TEMPST  
        ASL TEMPST  
        BICB #60,TIB ;CLEAR OFF ASCII  
        BISB TIB,TEMPST ;MOV CHAR TO TEMPST
```

Address	Offset	Label	Operation	Comments
3869	014156	005337	000726	
3870	014162	001756		
3871	014164	000710		
3872	014166	013777	000724	164520
3873	014174	000207		7\$:
3874	014176	010046		OUT:
3875	014200	010146		TTINTS:
3876	014202	004737	013700	
3877	014206	012601		
3878	014210	012600		
3879	014212	000002		
3880				
3881				
3882				
3883	014214	112737	000001	000730
3884	014222	112737	000006	000733
3885	014230	112737	000005	000734
3886	014236	010346		
3887	014240	010446		
3888	014242	010546		
3889	014244	113704	000733	
3890	014250	005404		
3891	014252	062704	000006	
3892	014256	110437	000732	
3893	014262	113704	000730	
3894	014266	016605	000010	
3895	014272	005003		
3896	014274	006105		1\$:
3897	014276	000404		
3898	014300	006105		2\$:
3899	014302	006105		
3900	014304	006105		
3901	014306	010503		
3902	014310	006103		3\$:
3903	014312	105337	000732	
3904	014316	100015		
3905	014320	042703	177770	
3906	014324	001002		
3907	014326	005704		
3908	014330	001403		
3909	014332	005204		4\$:
3910	014334	052703	000060	
3911	014340	105777	164252	5\$:
3912	014344	100375		
3913	014346	110377	164246	
3914	014352	105337	000734	7\$:
3915	014356	003350		
3916	014360	002402		
3917	014362	005204		
3918	014364	000745		
3919	014366	012605		6\$:
3920	014370	012604		
3921	014372	012603		
3922	014374	000207		
3923				

Operation	Comments
DEC	
BEQ	5\$
BR	1\$
MOV	TEMPST,@SR
RTS	PC
MOV	RO,-(SP)
MOV	R1,-(SP)
JSR	PC,CKSWR
MOV	(SP)+,R1
MOV	(SP)+,RO
RTI	
MOV	#1,FILL
MOV	#6,MODE+1
MOV	#5,CNT
MOV	R3,-(SP)
MOV	R4,-(SP)
MOV	R5,-(SP)
MOV	MODE+1,R4
NEG	R4
ADD	#6,R4
MOV	R4,MODE
MOV	FILL,R4
MOV	10(%6),R5
CLR	R3
ROL	R5
BR	3\$
ROL	R5
ROL	R5
ROL	R5
MOV	R5,R3
ROL	R3
DECB	MODE
BPL	7\$
BIC	#177770,R3
BNE	4\$
TST	R4
BEQ	5\$
INC	R4
BIS	#60,R3
TSTB	@TPS
BPL	5\$
MOV	R3,@TPB
DECB	CN-
BGT	2\$
BLT	6\$
INC	R4
BR	2\$
MOV	(SP)+,R5
MOV	(SP)+,R4
MOV	(SP)+,R3
RTS	PC

:*****

: ONLY WANT 6 NUMBERS AND <CR>
: IF = 7 TOO MANY NUMBERS
: GET NEXT CHAR
: CHANGE SWR CONTENTS
: RETURN TO PROGRAM
: INTERRUPT SERVICE ROUTINE
: SAVE R0 AND R1
: GO TO SUBR TO SERVICE TTY INTERRUPT
: RESTORE R1 AND R0

: RETURN FROM INTERRUPT

: SET THE ZERO FILL SWITCH
: SET FOR SIX(6) DIGITS
: SET THE ITERATION COUNT
: SAVE R3
: SAVE R4
: SAVE R5
: GET THE NUMBER OF DIGITS TO TYPE

: SUBTRACT IT FOR MAX. ALLOWED
: SAVE IT FOR USE
: GET THE ZERO FILL SWITCH
: PICKUP THE INPUT NUMBER
: CLEAR THE OUTPUT WORD
: ROTATE MSB INTO 'C'
: GO DO MSB
: FORM THIS DIGIT

: GET LSB OF THIS DIGIT
: TYPE THIS DIGIT?
: BR IF NO
: GET RID OF JUNK
: TEST FOR 0
: SUPPRESS THIS 0
: BR IF YES
: DON'T SUPPRESS ANYMORE 0'S
: MAKE THIS DIGIT ASCII
: IS PRINTER READY FOR CHARACTER?
: IF NOT, TRY AGAIN
: TYPE OUT NUMBER
: COUNT BY 1
: BR IF MORE TO DO
: BR IF DONE
: INSURE LAST DIGIT ISN'T A BLANK
: GO DO THE LAST DIGIT
: RESTORE R5
: RESTORE R4
: RESTORE R3
: RETURN FROM INTERRUPT PC

3924
3925
3926 014376 001007 007600 055103
3927 014404 040514 043103 020060
3928 014412 040514 033063 052040
3929 014420 051105 020115 042050
3930 014426 030514 020061 020046
3931 014434 046113 030461 100051
3932 014442 040514 033063 052040
3933 014450 051105 044515 040516
3934 014456 020114 044504 043501
3935 014464 047516 052123 041511
3936 014472 200
3937 014473 104 030514 020061
3938 014500 020046 046113 030461
3939 014506 044440 052116 051105
3940 014514 040506 042503 005200
3941 014522 000
3942 014523 200 042412 042116
3943 014530 047440 020106 040520
3944 014536 051523 020040
3945 014542 030060 030060 005200
3946 014550 000
3947 014551 200 047503 051516
3948 014556 046117 020105 020046
3949 014564 030060 042040 030514
3950 014572 023461 020123 047125
3951 014600 042504 020122 042524
3952 014606 052123 005200 000
3953 014613 007 100002 005017
3954 014620 042524 052123 021440
3955 014626 000
3956 014627 060 030060 041440
3957 014634 046117 046525 051516
3958 014642 005200 000
3959 014645 124 050131 020105
3960 014652 047101 020131 051120
3961 014660 047111 040524 046102
3962 014666 020105 044103 051101
3963 014674 041501 042524 020122
3964 014702 047101 020104 044514
3965 014710 052123 047105 043040
3966 014716 051117 041040 046105
3967 014724 027114 027056 027056
3968 014732 027056 027056 027056
3969 014740 027056 027056 000
3970 014745 200 047516 020124
3971 014752 047105 052517 044107
3972 014760 041440 046117 046525
3973 014766 051516 000200
3974 014772 042600 044103 020117
3975 015000 042524 052123 052040
3976 015006 051105 044515 040516
3977 015014 042524 100104 000
3978 015021 040 040
3979 015023 040 020040 040

.SBTTL MISC. DIAGNOSTIC MESSAGES
STARTM: .ASCII <7><2><ACRLF><17>/CZLACFO LA36 TERM (DL11 & KL11)/<ACRLF>
.ASCII /LA36 TERMINAL DIAGNOSTIC/<ACRLF>
.ASCIIZ /DL11 & KL11 INTERFACE/<ACRLF><12>
ENDPAS: .ASCII <ACRLF><12>/END OF PASS /
PASMES: .ASCIIZ /0000/<ACRLF><12>
DL11S: .ASCII <ACRLF>/CONSOLE & /
DL11S1: .ASCIIZ /00 DL11'S UNDER TEST/<ACRLF><12>
HDRMSG: .ASCIIZ <7><2><ACRLF><17><12>/TEST #/
HDRO: .ASCIIZ /000 COLUMNS/<ACRLF><12>
EO25MA: .ASCII /TYPE ANY PRINTABLE CHARACTER /
.ASCIIZ /AND LISTEN FOR BELL...../
EO25MB: .ASCIIZ <ACRLF>/NOT ENOUGH COLUMNS/<ACRLF>
EOEND: .ASCIIZ <ACRLF>/ECHO TEST TERMINATED/<ACRLF>
EO23M: .ASCII / /
LINES: .ASCII / / ;MSG FOR TEST E024

3980 015027 040 020040 020040
3981 015034 020040 100040 000
3982 015041 200 005017 042523
3983 015046 042514 052103 052040
3984 015054 051505 020124 052516
3985 015062 041115 051105 020040
3986 015070 000
3987 015071 105 042526 116
3988 015075 117 042104 040
3989 015101 124 050131 020105
3990 015106 047101 020131 044103
3991 015114 051101 041501 042524
3992 015122 000122
3993 015124 051525 020105 047523
3994 015132 052106 040527 042522
3995 015140 051440 044527 041524
3996 015146 020110 042522 020107
3997 015154 052101 046440 046505
3998 015162 051117 020131 042101
3999 015170 051104 030440 033067
4000 015176 000007
4001
4002
4003
4004 015200 020040 043536 000040
4005 015206 020040 053523 036522
4006 015214 000040
4007 015216 020040 042516 036527
4008 015224 000040
4009 015226 037440 000040
4010
4011 000001

LINESA: .ASCIZ / /<ACRLF>
MESG3: .ASCIZ <ACRLF><17><12>/SELECT TEST NUMBER /
EVEN: .ASCII /EVEN/
ODD: .ASCII /ODD /
OPMSG: .ASCIZ /TYPE ANY CHARACTER/
NOSWR: .ASCIZ /USE SOFTWARE SWITCH REG AT MEMORY ADDR 176/<7>
:CHGF6 *****
CNTG: .ASCIZ / ^G /
SWR: .ASCIZ / SWR /
NEW: .ASCIZ / NEW /
QUEST: .ASCIZ / ? /
:*****
.END

XCSR	000670	3239	3273	3290	3307	3444	3458	3467	3507	3509	3567	3586	3789	
		1376#	2197*	2198	2200	2205*	2260*	2267	2269*	2270	2317*	2318*	2321	2323
SAREAD	003726	2325*	2326*	2329*	2331*									
SBTASC	004074	1842	2116#	2126										
SCR	003314	1844	2165#											
SCRLF	003302	1843	1926#											
SFORWD	003650	1835	1921#											
SLF	003304	1845	2092#											
SPRHDR	003324	1837	1922#											
SPRNT	004402	1839	1936#											
SPRTC	004412	1840	2234#	2235										
SREAD	004200	1838	2248#											
SREADC	004272	1841	2193#	2217										
SSCRLF	003230	1846	2195	2209#	2210									
.	= 015232	1836	1894#											
		1298#	1300	1302	1304	1306	1308	1310	1314	1315#	1319#	1323#	1327#	1338#
		3059#	3774#											

. ABS. 015232 000

ERRORS DETECTED: 0

CZLACF.BIN,CZLACF.LST/CRF/SOL/NL:TOC-(CZLACF.P11
RUN-TIME: 12 25 1 SECONDS
RUN-TIME RATIO: 142/40=3.5
CORE USED: 7K (13 PAGES)