

.REPT 0

000583

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

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZKLA-E-D
PRODUCT NAME: KL11/DL11-A TELETYPE TESTS
DATE: 21-DECEMBER-1975
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: R. KOLLER
REVISED BY: AL LOSCHAK :SUPPORT SOFTWARE SWITCH REGISTER
THIS MAINDEC OBSOLETES MAINDEC-11-DZKLAD

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
1972,1975

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPLIED BY IT.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

MAIN, MACY11 27(657) 17-NOV-75 14:06 PAGE 1B
DZKLAE TABLE OF CONTENTS

1754	PRG0 = INPUT-OUTPUT LOGIC TESTS
2368	PRG1 READER TEST
2417	PRG2=PRINTER TESTS
2786	PRG3=PUNCH TEST
2883	PRG4=KEYBOARD TEST
2960	PRG5 COMBINED TEST
3108	PRG6, PRG7
3158	PRG10, PRG11, PRG12
3204	PRG13, PRG14

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

1. ABSTRACT

THIS MAINDEC CONSISTS OF A PACKAGE OF TEST PROGRAMS DESIGNED TO TEST AN ASR33, KSR33, ASR35, OR KSR35 TELETYPE WHEN ATTACHED TO A PDP11 SYSTEM THROUGH A KL11 OR DL11A TELETYPE CONTROL. ALL TESTS ARE INCLUDED IN A SINGLE OBJECT TAPE. NOTE: THE FOLLOWING PROGRAMING FORMAT IS ILLEGAL AND IS NOT USED IN THIS PROGRAM - MESSAGE, FILLER, FILLER, RESET, AND ANOTHER MESSAGE IMMEDIATELY.

THE AVAILABLE TEST PROGRAMS ARE LISTED HERE IN NUMERICAL ORDER:

PRG0-COMBINED INPUT-OUTPUT LOGIC TESTS
PRG1-READER TEST
PRG2-PRINTER TEST
PRG3-PUNCH TEST
PRG4-KEYBOARD TEST
PRG5-COMBINED READER-PUNCH-PRINTER TEST
PRG6-READER EXERCISER-SPECIAL BINARY COUNT PATTERN
PRG7-PRINTER EXERCISER
PRG10-SPECIAL BINARY COUNT PATTERN TAPE GENERATOR
PRG11-PUNCH CLOCK ADJUSTMENT ROUTINE
PRG12-READER CLOCK ADJUSTMENT ROUTINE
PRG13-MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
PRG14-MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN TEST.

PROGRAMS PRG0 THROUGH PRG5 ARE THE ACTUAL TELETYPE TESTS.
PROGRAMS PRG6 THROUGH PRG14 ARE UTILITY AND MAINTENANCE ROUTINES.

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11 SYSTEM. (4 K CORE).
B. ASR33, KSR33, ASR35, OR KSR35 TELETYPE WITH KL11 OR DL11A TTY CONTROL.

THE TELETYPE MUST HAVE STANDARD PERIPHERAL ADDRESSES.
REFER TO SECTION 7.3 IF THE TELETYPE DOES NOT HAVE STANDARD PERIPHERAL ADDRESSES.

2.2 STORAGE

THIS PROGRAM USES LOCATION 00000 THROUGH 16000.

3. LOADING PROCEDURE

THIS PROGRAM'S OBJECT TAPE IS PUNCHED IN ABSOLUTE FORMAT.
THE ABS LOADER IS USED TO LOAD THE PROGRAM.

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

4. USE PROCEDURE

THIS PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT A CONSOLE PROCESSOR, IF A CONSOLE MACHINE IS USED; THEN THE PROGRAM LOOKS AT THE HARDWARE SWITCH REGISTER, IF A CONSOLE=LESS MACHINE IS USED; THEN THE PROGRAM AUTOMATICALLY LOOKS AT THE CONTENTS OF LOCATION SOFTSR (176) AS A SWITCH REGISTER.

IT'S THE RESPONSIBILITY OF THE OPERATOR TO SET UP THIS LOCATION PRIOR TO STARTING THE PROGRAM.

IF IT HALTS THEN THE OPERATOR HAS TO SET UP THE SWITCH REGISTER AND RESTART AT THE NEXT LOCATION OF THE NORMAL HALT. IF A SWITCH REGISTER IS AVAILABLE ALL THAT THE OPERATOR HAS TO DO IS SET THE S,R. AND HIT CONTINUE.

4.1 TELETYPE IDENTIFICATION

BEFORE RUNNING ANY OF THE TEST PROGRAMS, DEPOSIT IN LOCATION 001230 ONE OF THE FOLLOWING NUMBERS:

000000 IF ASR33 IS ATTACHED, OR 000010 IF KSR33 IS ATTACHED, OR 000001 IF ASR35 IS ATTACHED, OR 000011 IF KSR35 IS ATTACHED,

TELETYPE IDENTIFICATION NEED BE DONE ONLY ONCE PER PROGRAM LOAD, UNLESS MORE THAN ONE TYPE OF TELETYPE IS ATTACHED TO SYSTEM.

4.2 PRG0 USE PROCEDURE (DESCRIPTION IN SECTION 8.1)

- A. IDENTIFY TELETYPE AS PER SECTION 4.1.
- B. SET TELETYPE TO ON=LINE.
- C. LOAD SPECIAL BINARY COUNT PATTERN TEST TAPE IN TAPE READER AND MAKE READER READY (DATA OVER PINS)
- D. LOAD ADDRESS 000200
- E. SET SR TO 000000. PRESS START
- F. THE PROGRAM STOPS AT COMMON HALT.
- G. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.

THIS PROGRAM'S SR OPTIONS ARE: (EXPLAINED IN SECTION 7.2)

SR15 HALT AT END OF ROUTINE
SR14 SCOPE MODE
SR11 INHIBIT ITERATION
SR10 LOOP PROGRAM
SR9 SELECT ROUTINE
SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.

H. PRESS CONTINUE. THE PROGRAM IS EXECUTED AND STOPS AT

.MAIN. MACY11 27(657) 17-NOV-75 14106 PAGE 4
DZKLAE

149
150
151
152

PROGRAM END HALT WHEN COMPLETED, PROVIDED NO ERRORS OCCUR.
I. REFER TO SECTION 6, IF ERRORS OCCUR.

EXECUTION TIME: NORMAL ERROR FREE PASS ABOUT 4 MINUTES.

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

4.3 PRG1 USE PROCEDURE (DESCRIPTION IN SECTION 8.2)

-
- A. IDENTIFY TELETYPE AS PER SECTION 4.1
 - B. SET TELETYPE TO ON-LINE
 - C. LOAD SPECIAL BINARY COUNT PATTERN TEST TAPE IN TAPE READER AND MAKE READER READY. DATA HOLES MUST BE OVER THE READ PINS.
 - D. LOAD ADDRESS 000200.
 - E. SET SR TO 000001, PRESS START
 - F. PROGRAM STOPS AT COMMON HALT.
 - G. SET ANY DESIRED SR OPTIONS, NORMAL RUN IS WITH SR = 000000.

THIS PROGRAM'S SR OPTIONS ARE: (EXPLAINED IN SECTION 7.2)

- SR15 HALT AT END OF ROUTINE
- SR11 INHIBIT ITERATION
- SR10 LOOP PROGRAM
- SR9 SELECT ROUTINE
- SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.

- H. PRESS CONTINUE. READER IS TESTED. PROGRAM HALTS AT PROGRAM END HALT IF NO ERRORS OCCUR.
- I. REFER TO SECTION 6. IF ERRORS OCCUR.

EXECUTION TIME: NORMAL ERROR FREE PASS ABOUT 11 MINUTES.

4.4 PRG2 USE PROCEDURE (DESCRIPTION IN SECTION 8.3)

-
- A. IDENTIFY TELETYPE AS PER SECTION 4.1
 - B. SET TELETYPE TO ON-LINE
 - C. LOAD ADDRESS 000200.
 - D. SET SR TO 000002, PRESS START
 - E. PROGRAM STOPS AT COMMON HALT.
 - F. SET ANY DESIRED SR OPTIONS, NORMAL RUN IS WITH SR = 000000.

THIS PROGRAM'S SR OPTIONS ARE: (EXPLAINED IN SECTION 7.2)

- SR15 HALT AT END OF ROUTINE
- SR10 LOOP PROGRAM
- SR9 SELECT ROUTINE
- SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.

- H. PRESS CONTINUE. THE TELEPRINTER WILL BE EXERCISED AND THE PROGRAM WILL STOP AT PROGRAM END HALT WHEN COMPLETED.
- I. ERROR DETECTION IS BY VISUAL INSPECTION OF RESULTING PRINTOUT.

EXECUTION TIME: NORMAL ERROR FREE PASS ABOUT 12 MINUTES.

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

4.5

PRG3 USE PROCEDURE (DESCRIPTION IN SECTION 8,4)

- A. IDENTIFY TELETYPE AS PER SECTION 4.1
- B. WITH TELETYPE OFF-LINE, PUNCH FIVE INCHES OF BLANK LEADER. SET TELETYPE TO ON-LINE.
- C. LOAD BLANK LEADER IN PAPER TAPE READER AND MAKE READER READY. PUNCH TO READER SLACK SHOULD NOT BE EXCESSIVE.
- D. LOAD ADDRESS 000200.
- E. SET SR TO 000003. PRESS START
- F. PROGRAM STOPS AT COMMON HALT.
- G. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.

THIS PROGRAM'S SR OPTIONS ARE: (EXPLAINED IN SECTION 7.2)

SR15 HALT AT END OF ROUTINE
SR11 INHIBIT ITERATION
SR10 LOOP PROGRAM
SR9 SELECT ROUTINE
SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.

- H. PRESS CONTINUE. THE PROGRAM WILL EXERCISE THE PUNCH AND VERIFY THE DATA PUNCHED.
- I. IF ANY ERRORS OCCUR REFER TO SECTION 6. ERRORS.

EXECUTION TIME: NORMAL ERROR FREE PASS ABOUT 40 MINUTES.

4.6

PRG4 USE PROCEDURE (DESCRIPTION IN SECTION 8,5)

- A. IDENTIFY TELETYPE AS PER SECTION 4.1.
- B. SET TELETYPE ON-LINE.
- C. LOAD ADDRESS 000200.
- D. SET SR TO 000004. PRESS START
- E. THE PROGRAM TYPES "KEYBOARD TEST" AND STOPS AT COMMON HALT.
- F. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.
THIS PROGRAM'S SR OPTIONS ARE: (EXPLAINED IN SECTION 7.2)

SR15 HALT AT END OF ROUTINE
SR10 LOOP PROGRAM
SR9 SELECT ROUTINE
SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.

- G. PRESS CONTINUE. FOLLOW TYPED INSTRUCTIONS. WHEN DONE PROGRAM STOPS AT PROGRAM END HALT.

EXECUTION TIME: PROGRAM IS USER DEPENDENT.

256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307

4.7

PRG5 USE PROCEDURE (DESCRIPTION IN SECTION 8.6)

- A. IDENTIFY TELETYPE AS PER SECTION 4.1.
- B. WITH TELETYPE OFF-LINE PUNCH FIVE INCHES OF BLANK LEADER.
SET TELETYPE TO ON-LINE.
- C. LOAD BLANK LEADER IN PAPER TAPE READER AND MAKE READER READY.
PUNCH TO READER SLACK SHOULD NOT BE EXCESSIVE.
- D. LOAD ADDRESS 000200.
- E. SET SR TO 000005. PRESS START.
- F. PROGRAM STOPS AT COMMON HALT.
- G. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.

THIS PROGRAM'S SR OPTIONS ARE: (EXPLAINED IN SECTION 7.2)
SR8 DISABLE STALL MODE AND RUN FULL SPEED
- H. PRESS CONTINUE. THE PRINTER, PUNCH, AND READER ARE EXERCISED IN
COMBINATION USING DATA CODES FROM 241 TO 337. THE PATTERN IS
ROTATED SO THAT EACH CHARACTER IS TYPED IN EVERY PRINT POSITION.
- I. REFER TO SECTION 6. ERRORS IF ERRORS OCCUR.

EXECUTION TIME: ONE 63 LINE PASS TAKES ABOUT 18 MINUTES.

4.8

PRG6 USE PROCEDURE (DESCRIPTION IN SECTION 8.7)

- A. IDENTIFY TELETYPE AS PER SECTION 4.1
- B. SET TELETYPE TO ON-LINE
- C. LOAD SPECIAL BINARY COUNT PATTERN TEST TAPE IN TAPE READER
AND MAKE READER READY. DATA HOLES MUST BE OVER READ PINS.
- D. LOAD ADDRESS 000200.
- E. SET SR TO 000006. PRESS START
- F. THE PROGRAM EXERCISES THE READER CONTINUOUSLY.
- G. SR8 DISABLE STALL MODE AND RUN FULL SPEED OPTION MAY BE
USED AT ANY TIME. SECTION 7.2 DESCRIBES SR OPTIONS.
- H. REFER TO SECTION 6. ERRORS, IF ERRORS OCCUR.

EXECUTION TIME: CONTINUOUS RUNNING.

4.9

PRG7 USE PROCEDURE (DESCRIPTION IN SECTION 8.8)

- A. IDENTIFY TELETYPE AS PER SECTION 4.1
- B. SET TELETYPE TO ON-LINE
- C. LOAD ADDRESS 000200
- D. SET SR TO 000007. PRESS START
- E. THE PROGRAM TYPES "TYPE IN DATA"
- F. KEY IN ANY FIVE CHARACTERS TO BE TYPED.
- G. KEY IN EITHER A RUBOUT FOR FULL SPEED TYPING, OR ANY
OTHER CHARACTER FOR RANDOM STALLS BETWEEN CHARACTERS.
- H. THE PROGRAM TYPES CONTINUOUSLY LINES CONTAINING THE
FIVE CHARACTERS SPECIFIED, UNTIL SR15 IS SET TO A 1. AT
THAT POINT THE PROGRAM GOES TO STEP E.

308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345

4.10 PRG10 USE PROCEDURE (DESCRIPTION IN SECTION 8.9)

- A. IDENTIFY TELETYPE AS PER SECTION 4.1
- B. SET TELETYPE TO ON-LINE
- C. TURN ON PUNCH
- D. LOAD ADDRESS 000200.
- E. SET SR TO 000010. PRESS START
- F. THE PROGRAM CONTINUOSLY PUNCHES THE SPECIAL BINARY COUNT COUNT PATTERN UNTIL STOPPED BY USER.

EXECUTION TIME: CONTINUOUS RUNNING.

4.11 PRG11 USE PROCEDURE (DESCRIPTION IN SECTION 8.10)

PRG11 IS USED AS AN AID IN ADJUSTING THE TELETYPE PUNCH CLOCK, AND IN OBSERVING THE DATA BITS AS THEY ARE SHIFTED OUT OF THE PUNCH BUFFER, A SCOPE IS REQUIRED.

TO ADJUST THE PUNCH CLOCK PROCEED AS FOLLOWS:

- A. LOAD ADDRESS 000200
- B. SET SR TO 000011. PRESS START.
- C. PROGRAM STOPS AT COMMON HALT.
- D. SET ANY DESIRED ASCII CODE IN LEFT HALF OF SR.
- E. SET NUMBER OF MILLISECONDOS TO DELAY BETWEEN PUNCH COMMANDS IN RIGHT HALF OF SR. THE NUMBER OF MILLISECONDOS SELECTED SHOULD BE LONG ENOUGH FOR THE ENTIRE PUNCH OPERATION TO COMPLETE, A SUGGESTED STARTING NUMBER IS 177.
- F. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY, FIRST IT LOADS THE PUNCH BUFFER WITH THE CHARACTER IN SR LEFT, AND THEN DELAYS FOR THE NUMBER OF MILLISECONDOS SPECIFIED IN SR RIGHT BEFORE RELOADING THE PUNCH BUFFER AGAIN.
- G. SET UP A SCOPE AND DISPLAY THE PUNCH CLOCK PULSES. ADJUST THE PUNCH CLOCK ACCORDING TO SPECIFICATIONS.

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

4.12 PRG12 USE PROCEDURE (DESCRIPTION IN SECTION 8.11)

PRG12 IS USED AS AN AID IN ADJUSTING THE READER/KYBD CLOCK, AND IN OBSERVING THE DATA BITS AS THEY ARE SHIFTED INTO THE READER/KYBD BUFFER, A SCOPE IS REQUIRED.

THE PROGRAM MAKES USE OF THE PUNCH MAINTENANCE BIT FEATURE IN ORDER TO CAUSE THE DATA OUTPUTTED TO THE PUNCH BUFFER TO BE SHIFTED INTO THE READER/KYBD BUFFER.

TO ADJUST THE READER CLOCK PROCEED AS FOLLOWS:

- A. LOAD ADDRESS 000200
- B. SET SR TO 000012. PRESS START.
- C. PROGRAM STOPS AT COMMON HALT.
- D. SET ANY DESIRED ASCII CODE IN LEFT HALF OF SR.
- E. SET NUMBER OF MILLISECONDS TO DELAY BETWEEN PUNCH COMMANDS IN RIGHT HALF OF SR. THE SELECTED NUMBER SHOULD BE LONG ENOUGH FOR THE ENTIRE PUNCH/READ OPERATION TO COMPLETE. A SUGGESTED STARTING NUMBER IS 177.
- F. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY. FIRST IT LOADS THE PUNCH BUFFER WITH THE CHARACTER IN SR LEFT, AND THEN DELAYS THE NUMBER OF MILLISECONDS SPECIFIED IN SR RIGHT. AS THE DATA BITS ARE SHIFTED OUT OF THE PUNCH BUFFER, THE READER CLOCK STARTS, AND THE DATA BITS ARE SHIFTED INTO THE READER BUFFER. AT THE END OF THE DELAY THE PROGRAM MOVES THE READ BUFFER CONTENTS TO REG 0, AND ISSUES 5 RESET INSTRUCTIONS IN ORDER TO MAKE THE READER BUFFER CONTENTS VISIBLE IN THE RIGHT HALF OF THE DATA LIGHTS.
- G. SET UP A SCOPE AND DISPLAY THE READ CLOCK PULSES. ADJUST THE READER CLOCK ACCORDING TO SPECIFICATIONS.

380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418

4.13 PRG13 USE PROCEDURE (DESCRIPTION IN SECTION 8.12)

- A. SET TELETYPE TO ON-LINE IF CHARACTER TO BE TESTED IS TO BE PUNCHED FOR VERIFICATION PURPOSES.
- B. TURN ON PUNCH IF CHARACTER IS TO BE PUNCHED.
- C. LOAD ADDRESS 000200.
- D. SET SR TO 000013. PRESS START
- E. THE PROGRAM STOPS AT COMMON HALT.
- F. SET CODE FOR CHARACTER TO BE TESTED IN THE LEFT HALF OF THE SR.
- G. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY, OUTPUTTING THE CHARACTER TO THE OUTPUT BUFFER AND CHECKING THAT THE READ BUFFER CONTAINS THE SAME CHARACTER WHEN THE READER DONE BIT BECOMES SET.
- H. REFER TO SECTION 6. ERRORS, IF ERRORS OCCUR.

EXECUTION TIME: CONTINUOUS RUNNING.

4.14 PRG14 USE PROCEDURE (DESCRIPTION IN SECTION 8.13)

- A. SET TELETYPE TO ON-LINE AND TURN ON PUNCH IF DATA IS TO BE PUNCHED FOR VERIFICATION PURPOSES.
- B. LOAD ADDRESS 000200.
- C. SET SR TO 000014. PRESS START
- D. THE PROGRAM RUNS CONTINUOUSLY. THE SPECIAL BINARY COUNT PATTERN IS OUTPUTTED TO THE OUTPUT BUFFER. EACH TIME THE READER DONE BIT BECOMES SET THE CHARACTER IN THE READ BUFFER IS CHECKED TO SEE THAT IT MATCHES THE PREVIOUSLY OUTPUTTED CHARACTER. THE PROGRAM STALLS RANDOMLY BETWEEN CHARACTERS. TO RUN AT FULL SPEED, SET SR8 TO A 1.
- E. REFER TO SECTION 6. ERRORS, IF ERRORS OCCUR.

EXECUTION TIME: CONTINUOUS RUNNING.

419	5.	PROGRAM AND/OR OPERATOR ACTION	
420		-----	
421			
422	5.1	NORMAL HALTS	
423		-----	
424			
425		LOC 001430	COMMON HALT. THIS HALT OCCURS WHENEVER
426			THE PROGRAM IS AWAITING USER INTERVENTION.
427			REGISTER ZERO CONTAIN THE ADDRESS OF
428			INSTRUCTION THAT GENERATED THE CALL TO THE
429			COMMON HALT.
430			THIS ADDRESS IS DISPLAYED IN THE DATA LIGHTS,
431			IF ANY.
432			
433		LOC 001524	END OF ROUTINE HALT. THIS HALT OCCURS AT THE
434			END OF A TEST ROUTINE IF SR15 IS SET TO A 1.
435			TO PROCEED, PRESS CONTINUE. PROGRAMS PRG0,
436			PRG1, PRG2, PRG3 AND PRG4 USE THE ROUTINE END
437			OPTION.
438			
439		LOC 2106	PROGRAM END HALT. THIS HALT NORMALLY OCCURS AT
440			THE END OF PROGRAMS PRG0, PRG1, PRG2, PRG3, AND
441			PRG4 UNLESS THE LOOP PROGRAM OPTION IS SET. (SR10)
442			
443	6.	ERRORS	
444		-----	
445			
446	6.1	ERROR HALTS	
447		-----	
448			
449		LOC 001442	UNCONDITIONAL ERROR HALT. REGISTER ZERO CONTAINS
450			ADDRESS OF INSTRUCTION THAT GENERATED THE
451			ERROR CALL. REFER TO PROGRAM LISTING.
452			THIS ADDRESS IS DISPLAYED IN THE DATA LIGHTS,
453			IF ANY.
454			
455		LOC 001504	CONDITIONAL ERROR HALT. THIS CALL WILL ALWAYS
456			OCCUR, UNLESS SR14 IS SET TO A 1. (SCOPE MODE)
457			REGISTER ZERO CONTAIN ADDRESS OF INSTRUCTION THAT
458			GENERATED ERROR CALL. REFER TO PROGRAM LISTING.
459			THIS ADDRESS IS DISPLAYED IN THE DATA LIGHTS,
460			IF ANY.
461			
462		LOC 001462	DATA ERROR HALT. OCCURS WHEN A PROGRAM OR
463			ROUTINE CHECKING DATA FINDS THAT THE EXPECTED
464			AND THE RECEIVED DATA DO NOT AGREE. THE LEFT
465			HALF OF REGISTER ZERO CONTAIN THE EXPECTED
466			8 BIT DATA. THE RIGHT HALF CONTAINS THE
467			RECEIVED 8 BIT DATA.
468			THIS DATA IS DISPLAYED IN THE DATA LIGHTS,
469			IF ANY.

470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519

6.2 NON RECOVERABLE ERROR HALTS

A NON-RECOVERABLE ERROR HALT WILL OCCUR AT THE ADDRESSES LISTED BELOW IF THROUGH HARDWARE OR SOFTWARE FAILURE, PROGRAM CONTROL IS TRANSFERRED TO AN UNEXPECTED AREA BETWEEN 000000 AND 000776.

- 000002 RESERVED AREA
- 000006 ERROR TRAP
- 000012 RESERVED INSTRUCTION TRAP
- 000016 DEBUG TRAP
- 000022 IOT TRAP
- 000026 POWER FAIL TRAP
- 000040 THROUGH 000176 SYSTEM SOFTWARE AND INTERRUPT VECTOR AREA, EXCEPT FOR KL11 INTERRUPT VECTORS.

TO FIND OUT WHERE THE PROGRAM WAS AT THE TIME THE FAILURE OCCURRED, PERFORM THE FOLLOWING STEPS:

- A. EXAMINE THE CONTENTS OF REGISTER 6 (ADDRESS 177706).
- B. TRANSFER THE CONTENTS OF REGISTER 6 TO THE SR, LOAD ADDRESS, AND EXAMINE.
- C. THE DATA SHOWN IN THE DATA LIGHTS IS THE VALUE OF THE PC WHEN THE FAILURE OCCURRED.
- D. LOCATE IN THE PROGRAM LISTING THE DISPLAYED PC VALUE.
- E. THE INSTRUCTION THAT IMMEDIATELY PREECEDES THE ONE REFERENCED BY THE DISPLAYED PC VALUE IS THE INSTRUCTION THAT WAS/WAS BEING EXECUTED WHEN THE FAILURE OCCURRED.

7. MISCELLANEOUS

7.1 TEST TAPES

MAINDEC-00-D2G4-PT SPECIAL BINARY COUNT PATTERN TEST TAPE IS RELEASED WITH THIS PROGRAM.

THE SPECIAL BINARY COUNT PATTERN TAPE IS PUNCHED WITH A PATTERN CONSISTING OF THE NUMBERS 000 THROUGH 377. EACH NUMBER IS IMMEDIATELY FOLLOWED BY ITS ONES COMPLEMENT NUMBER. FOR EXAMPLE:

001,376,002,375,003,374,004,373,ETC.

THE EASIEST WAY TO MAKE A SPECIAL BINARY COUNT PATTERN TEST LOOP IS TO OVERLAP THE TAPE AT THE POINT WHERE THE CHARACTERS 377,000,000,377 APPEAR. THAT SEQUENCE OF CHARACTERS APPEARS EVERY 512 CHARACTERS. THEREFORE, A MINIMUM SIZE TEST LOOP WOULD CONSIST OF 512 CHARACTERS.

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

7.2 SR OPTIONS

-
- SR15 - HALT AT END OF ROUTINE, FOR THOSE PROGRAMS CONSISTING OF A SET OF SEPARATE TEST ROUTINES, SR15 SET TO A 1 CAUSES THE PROGRAM TO HALT UPON COMPLETION OF THE ROUTINE CURRENTLY BEING EXECUTED. THREE POSSIBLE USES OF THIS OPTION ARE:
- A. TO STEP THROUGH A PROGRAM ONE ROUTINE AT A TIME.
 - B. WHEN AN UNPREDICTED FAILURE HAS OCCURRED (BLOW UP, HANG UP), TO ADVANCE THROUGH THE PROGRAM ONE ROUTINE AT A TIME UNTIL THE FAILURE OCCURS. THE ROUTINE FOLLOWING THE LAST IDENTIFIED ROUTINE WOULD BE THE FAILING ROUTINE.
 - C. WHEN A PROGRAM IS IN EXECUTION, TO DETERMINE HOW FAR THE PROGRAM HAS PROGRESSED.
- SR14 - SCOPE. THIS OPTION IS USED ONLY BY PRG0. THE OPTION CAUSES THE PROGRAM TO BYPASS ERROR HALTS, AND TO STAY IN THE FAILING ROUTINE.
- SR11 - INHIBIT ITERATION COUNT. THIS OPTION IS USED BY PRG0, PRG1, AND PRG3. THESE PROGRAMS CONSIST OF A SET OF ROUTINES EACH OF WHICH SPECIFIES THE NUMBER OF TIMES A TEST IS TO BE PERFORMED BY MEANS OF AN ITERATION COUNT. SETTING SR11 TO A 1 CAUSES THE PROGRAM TO DISREGARD THE ITERATION COUNT AND PERFORM THE TEST ONLY ONCE FOR EACH ROUTINE. TWO POSSIBLE USES OF THIS OPTION ARE:
- A. QUICK PASS. THE USER MAY ELECT TO RUN THROUGH A PROGRAM QUICKLY TO FIND OUT IF ANY FAILURES SHOW IMMEDIATELY. A SUCCESSFUL QUICK PASS HOWEVER, DOES NOT GUARANTEE THAT THE SAME PROGRAM WILL RUN ERROR-FREE WHEN PERFORMING A NORMAL ITERATION PASS.
 - B. SKIP OVER FAILING ROUTINE. WHEN A ROUTINE HAS DETECTED A SOLID FAILURE, THE ERROR WILL BE REPORTED MANY TIMES. TO GO ON TO THE NEXT ROUTINE, THE USER CAN INHIBIT ITERATION. IT WILL BE NECESSARY TO CAUSE THE PROGRAM TO STOP AT THE END OF THE ROUTINE BY SETTING SR15 TO A 1. OTHERWISE THE PROGRAM WOULD QUICKLY RUN THROUGH THE NEXT ROUTINE(S) ALSO.

566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618

(7.2 CONT'D)

- SR10 = LOOP PROGRAM. THIS OPTION IS USED BY PROGRAMS PRG0, PRG1, PRG2, PRG3, AND PRG4. SETTING SR10 TO A 1 CAUSES THE PROGRAM TO REPEAT ITSELF, INSTEAD OF STOPPING AT PROGRAM END HALT.
- SR9 = SELECT ROUTINE. THIS OPTION IS USED BY PROGRAMS PRG0, PRG1, PRG2, PRG3, AND PRG4. THE USER MAY ELECT TO RUN ONLY ONE SPECIFIC ROUTINE BY SETTING SR9 TO A 1, AND SR6 THROUGH SR8 TO THE NUMBER OF THE DESIRED ROUTINE. REFER TO THE INDIVIDUAL PROGRAM DESCRIPTION IN SECTION 8 TO OBTAIN THE ROUTINE NUMBER. THE ROUTINE NUMBER SELECTED MUST BE A VALID NUMBER, OR AN ERROR HALT WILL OCCUR. THE SELECT ROUTINE OPTION WILL BE HONORED BY THE PROGRAM UPON COMPLETION OF THE CURRENT ROUTINE, OR UPON STARTING THE PROGRAM.
- SR8 = DISABLE STALL MODE AND RUN FULL SPEED. USED BY PROGRAMS PRG5, PRG6, AND PRG14. THESE PROGRAMS OPERATE NORMALLY IN STALL MODE (TESTS OR EXERCISES ARE NOT FULL SPEED, BUT RANDOM DURATION DELAYS ARE INTRODUCED). SETTING SR8 TO A 1 CAUSES THE PROGRAMS TO PERFORM THEIR TESTS AT FULL SPEED.

7.3

TESTING KL11 AT NON-STANDARD ADDRESSES AND/OR VECTORS

THIS PROGRAM CAN TEST A KL11 ASSIGNED TO NON-STANDARD ADDRESSES AND VECTORS IF THOSE ADDRESSES ARE PROVIDED TO THE PROGRAM AS FOLLOWS:

- A. IMMEDIATELY AFTER LOADING THE PROGRAM CHANGE THE FOLLOWING LOCATIONS. REFER TO PROGRAM LISTING.

LOCATION	FROM STANDARD	TO NON-STANDARD
001210	177560	RDR/KYBD CSR ADDRESS
001212	177562	RDR/KYBD BUFFER ADDRESS
001214	177564	PCH/PRINTER CSR ADDRESS
001216	177566	PCH/PRINTER BUFFER ADDRESS
001220	000860	RDR/KYBD VECTOR ADDRESS
001222	000200	RDR/KYBD PRIORITY LEVEL
001224	000864	PCH/PRINTER VECTOR ADDRESS
001226	000200	PCH/PRINTER PRIORITY LEVEL

- B. PROCEED TO USE PROGRAM, OR
- C. DUMP OUT ENTIRE PROGRAM IN ABSOLUTE FORMAT, TO HAVE AN UPDATED OBJECT TAPE THAT REFLECTS YOUR SYSTEM, OR
- D. DUMP OUT ONLY LOCATIONS 001204 THROUGH 001222, AND SPLICE THE TAPE TO THE END OF THE STANDARD OBJECT TAPE. THIS PROCEDURE WOULD REQUIRE THAT THE SHORT LENGTH OF TAPE BE LOADED IMMEDIATELY AFTER THE MAIN PROGRAM, IN ORDER TO OVERLAY LOCATIONS 001204 THROUGH 001222.

619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670

8. DESCRIPTION

8.1 PRG0 PROGRAM DESCRIPTION

PRG0 TESTS THE INPUT AND OUTPUT LOGIC IN ONE PROGRAM. THE PROGRAM
CONSISTS OF 36 TEST ROUTINES NUMBERED FROM 00 TO 43(8).

RTN0 TESTS ABILITY TO REFERENCE THE READER/KYBD STATUS WORD (TKS)
WITHOUT TRAPPING.
RTN1 TESTS ABILITY TO REFERENCE THE READER/KYBD BUFFER (TKB)
WITHOUT TRAPPING.
RTN2 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH STATUS WORD (TPS)
WITHOUT TRAPPING.
RTN3 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH BUFFER (TPB)
WITHOUT TRAPPING.
RTN4 TESTS ABILITY TO SET AND CLEAR THE READER/KYBD IE BIT.
RTN5 CHECKS THAT READER /KYBD ID BIT CAN BE CLEARED WITH RESET INSTRUCTION.
RTN6 CHECKS THAT READER DONE BIT SETS SOMETIME AFTER READER ENABLE.
RTN7 TESTS THAT DONE BIT CAN BE READ RELIABLY.
RTN10 TESTS THAT RESET CLEARS DONE BIT.
RTN11 CHECKS THAT REFERENCING READER BUFFER CLEARS DONE BIT.
RTN12 TESTS THAT BUSY BIT BIT SETS SOMETIME BEFORE DONE SETS.
RTN13 TESTS THAT READER DONE BIT IS CLEARED BY START BIT. (WHEN
BUSY BECOMES SET.)
RTN14 TESTS THAT READ BUFFER CAN BE READ RELIABLY.
RTN15 CHECKS THAT READER DONE BIT IS ABLE TO CAUSE AN INTERRUPT. IF THE
INTERRUPT IS SERVICED, IT WILL HAVE OCCURRED AT CORRECT VECTOR.
RTN16 TESTS THAT READER DONE DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR
IS AT THE SAME PRIORITY AS THE READER'S INTERRUPT REQUEST LEVEL.
RTN17 TESTS THAT READER DONE CAUSES INTERRUPT WHEN THE PROCESSOR IS AT A
PRIORITY ONE LEVEL LOWER THAN THE READER'S INTERRUPT REQUEST LEVEL.
RTN20 CHECKS THAT READER DONE DOES NOT REINTERRUPT AFTER RTI
INSTRUCTION WHEN DONE BIT IS LEFT SET.
RTN21 TESTS ABILITY TO SET AND CLEAR THE PUNCH ID BIT.
RTN22 CHECKS THAT PUNCH ID BIT CAN BE CLEARED WITH RESET INSTRUCTION.
RTN23 TESTS ABILITY TO SET AND CLEAR PUNCH MAINTENANCE BIT.
RTN24 CHECKS THAT RESET INSTRUCTION CLEARS THE MAINTENANCE BIT.
RTN25 TESTS THAT RESET SETS THE PUNCH READY BIT, AND THAT THE
READY BIT CAN BE READ RELIABLY.
RTN26 TESTS THAT PUNCH READY IS CLEARED BY LOADING THE PUNCH BUFFER.
RTN27 TESTS THAT BYTE LOADING PUNCH BUFFER+1 DOES NOT CLEAR THE
PUNCH READY BIT.
RTN30 CHECKS THAT THE PUNCH BECOMES READY SOMETIME AFTER BUFFER LOAD.
RTN31 CHECKS THAT PUNCH READY BIT CAN CAUSE INTERRUPT, IF THE INTERRUPT
IS SERVICED, IT WILL HAVE OCCURRED AT THE CORRECT VECTOR
RTN32 TESTS THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WHEN THE
PROCESSOR IS AT A PRIORITY AS THE READER'S INTERRUPT REQUEST
LEVEL.
RTN33 TESTS THAT PUNCH READY CAUSES AN INTERRUPT WHEN THE PROCESSOR

671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711

8.1 (CONT'D)

IS AT PRIORITY ONE LEVEL LOWER THAN THE PUNCH INTERRUPT
REQUEST LEVEL.
RTN34 TESTS THAT READY DOES NOT REINTERRUPT AFTER RTI WITH READY SET.
RTN35 CHECKS THAT PUNCH READY CAUSES AN INTERRUPT IMMEDIATELY UPON
LOWERING PROCESSOR PRIORITY TO 0.
RTN36 CHECKS FOR CORRECT OPERATION OF WAIT INSTRUCTION. (REFER TO
PROGRAM LISTING).
RTN37 TESTS THAT LOADING PUNCH BUFFER WITH MAINTENANCE BIT SET
CAUSES READER DONE BIT TO SET SOMETIME AFTER.
RTN40 TESTS THAT CLEARING READY/INTERRUPT ENABLE CLEARS PUNCH
INTERRUPT REQUEST.
RTN41 TESTS THAT CLEARING READER DONE/INTERRUPT ENABLE CLEARS
READER INTERRUPT REQUEST.
RTN42 TESTS THAT THE DL11A,B KEYBOARD JUMPERS ARE CUT PROPERLY
RTN43 TESTS THAT THE DL11A,B PRINTER JUMPERS ARE CUT PROPERLY

8.2

PRG1 PROGRAM DESCRIPTION

PRG1 CHECKS FOR CORRECT AND RELIABLE OPERATION OF THE PAPER TAPE
READER. THE PROGRAM CONTAINS 3 ROUTINES NUMBERED FROM 00 TO 02.
RTN0 READS AND CHECKS 2000 CHARACTERS OF SPECIAL BINARY COUNT
PATTERN AT FULL SPEED.
RTN1 READS AND CHECKS 1000 CHARACTERS OF SPECIAL BINARY COUNT
PATTERN WITH RANDOM DURATION STALLS BETWEEN CHARACTERS.
RTN2 READS AND CHECKS 200 GROUPS OF CHARACTERS OF SPECIAL BINARY
COUNT PATTERN. LENGTH OF EACH GROUP IS RANDOM, BUT DOES NOT
EXCEED 15 CHARACTERS.

712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761

8.3 PRG2 PROGRAM DESCRIPTION

PRG2 EXERCISES THE TELEPRINTER FUNCTIONS. VERIFICATION OF TELEPRINTER OPERATION IS PERFORMED VISUALLY BY USER. THE PROGRAM CONTAINS 29 ROUTINES NUMBERED FROM 00 TO 34(0).

RTN0 CARRIAGE RETURN TEST, CHECKS THAT CARRIAGE CAN CORRECTLY RETURN TO PRINT POSITION 0 FROM EVERY OTHER PRINT POSITION.

RTN1 RIGHT MARGIN TEST. THIS ROUTINE VERIFIES THAT THE RIGHT MARGIN IS CORRECTLY SET FOR 72 PRINT POSITIONS. THE TEST TYPES 73 CHARACTERS. IF THE RIGHT MARGIN IS CORRECTLY SET, CHARACTER 73 SHOULD OVERPRINT CHARACTER 72. THE TYPED LINE SHOULD LOOK AS FOLLOWS:

---I---I---I---I---I---I---I---I---I---I---I---I---I---I---I---I---I

RTN2 SPACE TEST. CHECKS ABILITY OF TELEPRINTER TO SPACE CORRECTLY. THE TEST FIRST PRINTS REVERSE SLASHES (\) IN ALTERNATE PRINT POSITIONS, AND THEN SPACES FROM PRINT POSITION 0 TO EACH PRINT POSITION AND PRINTS A SLASH (/). THE TYPEOUT SHOULD LOOK AS FOLLOWS:

\/

RTN3 LINE FEED TEST, TESTS FOR ABILITY TO CORRECTLY PERFORM A LINE FEED. A RANDOM STALL OCCURS BETWEEN EACH LINE FEED. A CORRECTLY PERFORMED TEST WILL APPEAR AS DIAGONAL LINE BETWEEN PRINT POSITION 0 AND PRINT POSITION 72.

RTN4 TAB TEST. ASR/KSR35 ONLY. CHECKS ABILITY OF TELEPRINTER TO TAB TO TO A TAB POSITION FROM 1 TO 7 POSITIONS AWAY FROM THE TAB. THE FIRST LINE TYPED MARKS THE TAB POSITIONS. SUBSEQUENT LINES ARE THE ACTUAL TAB TEST. THE TYPEOUT LOOKS AS FOLLOWS:

/ / / / / / / / / /
/ / / / / / / / / /
/ / / / / / / / / /
/ / / / / / / / / /
/ / / / / / / / / /

(8,3 CONT'D)

762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797

RTN5 TYPES LINE OF CHARACTERS ABC
RTN6 TYPES LINE OF CHARACTERS DEF
RTN7 TYPES LINE OF CHARACTERS GHI
RTN10 TYPES LINE OF CHARACTERS JKL
RTN11 TYPES LINE OF CHARACTERS MNO
RTN12 TYPES LINE OF CHARACTERS PQR
RTN13 TYPES LINE OF CHARACTERS STU
RTN14 TYPES LINE OF CHARACTERS VWX
RTN15 TYPES LINE OF CHARACTERS YZ0
RTN16 TYPES LINE OF CHARACTERS 123
RTN17 TYPES LINE OF CHARACTERS 456
RTN20 TYPES LINE OF CHARACTERS 789
RTN21 TYPES LINE OF CHARACTERS !"#
RTN22 TYPES LINE OF CHARACTERS \$%&
RTN23 TYPES LINE OF CHARACTERS '()
RTN24 TYPES LINE OF CHARACTERS *+,
RTN25 TYPES LINE OF CHARACTERS -. /
RTN26 TYPES LINE OF CHARACTERS :; <
RTN27 TYPES LINE OF CHARACTERS => ?
RTN30 TYPES LINE OF CHARACTERS @[\
RTN31 TYPES LINE OF CHARACTERS]^ AND LEFT ARROW
RTN32 TYPES 2 LINES OF ALL CHARACTERS, FIRST LINE IS TYPED AT
FULL SPEED, SECOND LINE IS TYPED WITH RANDOM STALLS.
RTN33 TYPES 12 LINES OF ASR33 WORST CASE PATTERN, EVERY OTHER
LINE IS TYPED WITH RANDOM STALLS, THE ASR33 WORST CASE
PATTERN IS ' _W/W_
RTN34 TYPES 12 LINES OF ASR35 WORST CASE PATTERN, EVERY OTHER
LINE IS TYPED WITH RANDOM STALLS, THE ASR35 WORST CASE
PATTERN IS '[?C?['

798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841

8.4 PRG3 PROGRAM DESCRIPTION

PRG3 CHECKS FOR CORRECT PUNCH OPERATION. THE PROGRAM CONTAINS 3 ROUTINES NUMBERED FROM 00 TO 02.

RTN0 PUNCHES 512 CHARACTERS OF SPECIAL BINARY COUNT PATTERN AT FULL SPEED, AND VERIFIES THE PUNCHED DATA WITH THE PAPER TAPE READER, TEST IS DONE 5 TIMES.

RTN1 PUNCHES 512 CHARACTERS OF SPECIAL BINARY COUNT PATTERN WITH RANDOM DURATION STALLS BETWEEN EACH CHARACTER PUNCHED. THE DATA PUNCHED IS VERIFIED WITH THE PAPER TAPE READER, THE TEST IS DONE 5 TIMES.

RTN2 PUNCHES 512 CHARACTERS OF SPECIAL BINARY COUNT PATTERN IN GROUPS OF RANDOM LENGTH (15 CHARACTERS MAXIMUM). AFTER EACH GROUP IS PUNCHED A RANDOM DURATION STALL OCCURS, THE DATA PUNCHED IS VERIFIED WITH THE PAPER TAPE READER, THE TEST IS DONE 5 TIMES.

8.5 PRG4 PROGRAM DESCRIPTION

PRG4 IS USED TO TEST THE TELETYPE KEYBOARD. THE PROGRAM CONTAINS 3 ROUTINES NUMBERED FROM 00 TO 02.

RTN0 TESTS THAT TELETYPE CONTROL RESPONDS WHEN USER DEPRESSES A KEYBOARD KEY.

RTN1 ECHO TEST. THE TEST ECHOES ONTO THE TELEPRINTER THE CHARACTER RECEIVED FROM THE KEYBOARD. WHEN THE TEST SENSES A RUBOUT CHARACTER THE TEST IS ENDED. THE TEST ENABLES THE USER TO DETERMINE IF ALL PRINTABLE CODES CAN BE SUCCESSFULLY SENT TO THE TELETYPE CONTROL.

RTN2 OCTAL EQUIVALENT TEST, THE OCTAL EQUIVALENT OF ANY CHARACTER RECEIVED BY THE CONTROL IS TYPED, SENSING A RUBOUT ENDS THE TEST. THIS TEST ENABLES THE USER TO DETERMINE THAT ALL CODES INCLUDING NON-PRINTABLE CONTROL CODES ARE BEING CORRECTLY SENT TO THE TELETYPE CONTROL.

842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892

8.6 PRG5 PROGRAM DESCRIPTION

PRG5 EXERCISES THE PAPER TAPE READER, PAPER TAPE PUNCH, AND THE TELEPRINTER IN COMBINATION. ITS PURPOSE IS TO SHOW UP ANY FAILURES THAT MAY OCCUR AS A RESULT OF INTERACTION BETWEEN THE TELETYPE DEVICES AND THE PROCESSOR.

PRG5 IS A CONTINUOUS RUNNING PROGRAM. DATA CODES 241 THROUGH 337 ARE USED TO EXERCISE THE DEVICES. THE DATA IS OUTPUTTED TO THE PRINTER/PUNCH IN GROUPS OF 72 CHARACTERS PRECEDED BY CARRIAGE RETURN AND LINE FEED. ONE PROGRAM PASS CONSISTS OF 63 LINES FOLLOWED BY A BLANK LINE. THE DATA IS OUTPUTTED SO THAT EVERY CHARACTER WILL HAVE PRINTED IN EVERY PRINT POSITION BY THE TIME ONE PROGRAM PASS HAS BEEN COMPLETED.

THE PROGRAM WORKS IN THE INTERRUPT MODE, AND IS ARRANGED SO THAT THE PAPER TAPE READER WILL AT NO TIME BE MORE THAN 40 CHARACTERS BEHIND THE PUNCH (EXCLUDING INITIAL PUNCH TO READER SLACK.)

NORMAL OPERATION IS WITH RANDOM STALLS. SR8 MAY BE SET TO A 1 TO RUN THE TEST AT FULL SPEED.

8.7 PRG6 PROGRAM DESCRIPTION

PRG6 IS A CONTINUOUS RUNNING PROGRAM THAT EXERCISES THE PAPER TAPE READER USING A SPECIAL BINARY COUNT PATTERN TEST TAPE. THE PROGRAM STALLS RANDOMLY AFTER READING RANDOM LENGTH CHARACTER GROUPS (15 CHARACTERS MAXIMUM) AT FULL SPEED. SR8 MAY BE SET TO A 1 TO EXERCISE THE READER AT FULL SPEED.

8.8 PRG7 PROGRAM DESCRIPTION

PRG7 IS A PRINTER EXERCISER DESIGNED AS AN AID IN MAKING TELEPRINTER ADJUSTMENTS. THE PROGRAM PERMITS THE USER TO TYPE IN FIVE TEST CHARACTERS AND ONE FINAL CHARACTER THAT SIGNIFIES WHETHER FULL SPEED OR STALL OPERATION IS DESIRED. THE PROGRAM THEN TYPES LINES CONTAINING THE FIVE SELECTED CHARACTERS. WHEN THE USER WISHES TO CHANGE THE TEST CHARACTERS SR15 IS SET TO A 1. THE PROGRAM TERMINATES TYPING THE LINE BEFORE ACCEPTING NEW DATA.

8.9 PRG10 PROGRAM DESCRIPTION

PRG10 IS A TAPE GENERATOR THAT PUNCHES OUT THE SPECIAL BINARY COUNT PATTERN CONTINUOUSLY.

893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944

8.10 PRG11 PROGRAM DESCRIPTION

PRG11 IS USED AS AN AID IN ADJUSTING THE TELETYPE PUNCH CLOCK WITH THE AID OF A SCOPE. THE PROGRAM PERFORMS THE FOLLOWING SEQUENCE:

- A. LOAD PUNCH BUFFER WITH ASCII CODE IN SR LEFT.
- B. DELAY NUMBER OF MILLISECONDS SET IN SR RIGHT.
- C. GO TO STEP A.

8.11 PRG12 PROGRAM DESCRIPTION

PRG12 IS USED AS AN AID IN ADJUSTING THE TELETYPE READER CLOCK. A SCOPE IS REQUIRED. THE PROGRAM PERFORMS THE FOLLOWING SEQUENCE:

- A. SET PUNCH MAINTENANCE BIT.
- B. LOAD PUNCH BUFFER WITH CODE IN SR LEFT.
- C. DELAY NUMBER OF MILLISECONDS SET IN SR RIGHT.
- D. MOVE CONTENTS OF READ BUFFER TO REGISTER 0.
- E. ISSUE 5 RESET INSTRUCTIONS TO "FIX" READ BUFFER CONTENTS IN RIGHT HALF OF DATA LIGHTS.
- F. GO TO STEP A.

8.12 PRG13 PROGRAM DESCRIPTION

USING THE PUNCH MAINTENANCE BIT FEATURE, PRG13 TAKES THE ASCII CODE SET IN SR LEFT AND USES IT TO CHECK THE ABILITY OF THE CONTROL TO OUTPUT AND RECEIVE DATA. THE PROGRAM PERFORMS THE FOLLOWING SEQUENCE:

- A. SET PUNCH MAINTENANCE BIT.
- B. LOAD PUNCH BUFFER WITH CODE IN SR LEFT.
- C. WHEN READER DONE BIT SETS, COMPARE CODE IN SR LEFT WITH DATA IN READER BUFFER. HALT IF NOT SAME.
- D. WAIT FOR PUNCH DONE BIT TO SET AND GO TO STEP B.

8.14 PRG14 PROGRAM DESCRIPTION

USING THE PUNCH MAINTENANCE BIT FEATURE PRG14 USES THE SPECIAL BINARY COUNT PATTERN TO CHECK ABILITY OF THE CONTROL TO OUTPUT AND RECEIVE DATA. THE PROGRAM PERFORMS THE FOLLOWING STEPS:

- A. INITIALIZE BINARY COUNT PATTERN.
- B. SET PUNCH MAINTENANCE BIT.
- C. LOAD PUNCH BUFFER WITH BINARY COUNT CHARACTER.
- D. WHEN READER DONE BIT SETS, COMPARE BINARY CHARACTER WITH DATA IN READ BUFFER. HALT IF NOT SAME.
- E. WAIT FOR PUNCH DONE BIT TO SET AND GO TO STEP C.

.ENDR

945
 946
 947
 948
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 990
 991
 992
 993
 994
 995
 996
 997
 998

000000
 000002
 000002
 000004
 000006
 000010
 000012
 000014
 000016
 000020
 000022
 000024
 000026
 000026
 000030
 000032
 000034
 000036
 000046
 002062
 177776
 177776
 001200
 000240

```

.ABS
.LIST ME
.NLIST MD,MC
)PRG0= COMBINED INPUT-OUTPUT LOGIC TESTS.
)PRG1= READER TEST.
)PRG2= PRINTER TEST.
)PRG3= PUNCH TEST.
)PRG4= KEYBOARD TEST.
)PRG5= COMBINED TEST.
)PRG6= READER EXERCISER, SPECIAL BINARY COUNT PATTERN.
)PRG7= PRINTER EXERCISER.
)PRG10= SPECIAL BINARY COUNT PATTERN TAPE GENERATOR.
)PRG11= PUNCH CLOCK ADJUSTMENT ROUTINE.
)PRG12= READER CLOCK ADJUSTMENT ROUTINE.
)PRG13= MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
)PRG14= MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST.
)
)STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
)
)SR 15 = HALT AT END OF ROUTINE.
)SR 14 = SCOPE.
)SR 11 = INHIBIT ITERATION.
)SR 10 = LOOP PROGRAM
)SR 9 = SELECT ROUTINE.
)SR 8 = DISABLE STALL MODE AND RUN FULL SPEED.
)SR 6 THROUGH SR 0 = NUMBER OF ROUTINE TO BE SELECTED.
)
)0
)+2 )UNASSIGNED TRAP
)HALT
MACHERR: )+2 )SP OVERFLOW, BUS ERROR TRAP
)HALT
)+2 )RESERVED INSTRUCTION TRAP
)HALT
)+2 )TRACE TRAP
)HALT
)+2 )TRAP TO CALL IOX
)HALT
)+2 )POWER FAIL TRAP
)HALT
)EMTINT )EMT TRAP
)PRTY7
)DLYX )TRAP TRAP. SIMILAR TO EMT
)PRTY7
)LOC 40 THROUGH 776 ARE FILLED WITH .+2 AND HALT.
)
)=46
LOGIC )SET TO ADDRESS OF ACT 11 HOOKS
)EQUATE STATEMENTS
CC=177776
PSW=177776
SPBOT=1200
NOP=240

```

999	000000		
1000	100000		
1001	000000		
1002	000006		
1003	000007		
1004	100000		
1005	040000		
1006	020000		
1007	010000		
1008	004000		
1009	002000		
1010	001000		
1011	000400		
1012	000200		
1013	000100		
1014	000040		
1015	000020		
1016	000010		
1017	000004		
1018	000002		
1019	000001		
1020	005726		
1021	022626		
1022	000340		
1023	000300		
1024	000240		
1025	000200		
1026	000140		
1027	000100		
1028	000040		
1029	000000		
1030	000000		
1031	104400		
1032	000007		
1033	015142		
1034	015144		
1035	015254		
1036	015265		
1037	015256		
1038	015267		
1039	015366		
1040	015377		
1041	177777		
1042	177777		
1043	177777		
1044	177777		
1045	177777		
1046	000174		
1047	000174	177570	
1048	000176	000000	
1049		000200	
1050	000200	012706	001200
1051	000204	016746	177576
1052	000210	016746	177570

SRPTR: 177570
SOFTSR: 000000

OPEN=0
MANUAL=BIT15
R0=X0
R6=X6
PC=X7
BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1
POPSP=5726
POPSP2=022626
PRTY7=340
PRTY6=300
PRTY5=240
PRTY4=200
PRTY3=140
PRTY2=100
PRTY1=40
PRTY0=0
EMTX=0
DELAYX=TRAP+0
BELL=007
BLOCKA=DEND
BLOCK1=BLOCKA+2
BLOCKB=BLOCKA+112
BLKBB=BLOCKA+123
BLOCK2=BLOCKA+114
BLK2=BLOCKA+125
BLOCKC=BLOCKA+224
BLKCC=BLOCKA+235
P0TLST=-1
P1TLST=-1
P2TLST=-1
P3TLST=-1
P4TLST=-1
.=174
MOV #SPBOT,X6
MOV 6,=(SP)
MOV 4,=(SP)

JPOP THE STACK, SAME AS TST (6)+
JPOP STACK TWICE, SAME AS CMP (6)+,(6)+
JPRIORITY LEVEL DEFINITIONS

JSET BOTTOM OF THE STACK
JSAVE CURRENT VECTOR

1053	000214	012767	000230	177562	MOV	#18,4	ISET UP TIME OUT VECTOR
1054	000222	005777	177746	TST	#8RPTR		I TRY TO REFERENCE THE
1055							I HARDWARE SWITCH REGISTER
1056	000226	000404			BR	28	I BRANCH IF NO TIME OUT TRAP OCCURS
1057	000230	012767	000176	177736	181	MOV	I CHANGE THE SWITCH REGISTER POINTER
1058							I TO POINT TO A SOFTWARE SWITCH REGISTER
1059	000236	022626			CMP	(6)+,(6)+	I RESTORE THE STACK
1060	000240	012667	177540		MOV	(6)+,4	I RESTORE TIME OUT VECTOR
1061	000244	012667	177536		MOV	(6)+,6	
1062	000250	000167	001254		JMP	START	I GO TO START OF PROGRAM.
1063		001210					
1064	001210	177560			TKB1	177560	I LSR CSR
1065	001212	177562			TKD1	177562	I LSR BUFFER
1066	001214	177564			TPD1	177564	I LSP CSR
1067	001216	177566			TPB1	177566	I LSP BUFFER
1068	001220	000060			TKVTR1	60	I LSR INTERRUPT VECTOR
1069	001222	000200			TKLV1	PRTY4	I LSR PRIORITY LEVEL
1070	001224	000064			TPVTR1	64	I LSP INTERRUPT VECTOR
1071	001226	000200			TKLV1	PRTY4	I LSP PRIORITY LEVEL
1072	001230	000000			TTYTYP1	OPEN	
1073	001232	000000			PRGNUM1	OPEN	I CONTAINS CURRENT PROGRAM#
1074	001234	000000			KSTART1	OPEN	I CURRENT PROGRAM START ADDRESS.
1075	001236	000000			CURTST1	OPEN	I CONTAINS ADDR OF CURRENT TEST.
1076	001240	000000			RTNND1	OPEN	I CONTAINS CURRENT TEST #.
1077	001242	000000			NXTST1	OPEN	I CONTAINS ADDR OF NEXT TEST.
1078	001244	000000			ICTR1	OPEN	I CONTAINS CURRENT ITERATION COUNT
1079	001246	000000			SCOPTR1	OPEN	I CONTAINS CURRENT SCOPE POINTER.
1080	001250	000000			PRGID1	OPEN	I CONTAINS PROGRAM INDICATORS
1081	001252	005144			PRGTAB1	PRG0	I PRG0 START ADDRESS
1082	001254	007652				PRG1	I PRG1 START ADDRESS
1083	001256	010026				PRG2	I PRG2 START ADDRESS
1084	001260	011462				PRG3	I PRG3 START ADDRESS
1085	001262	012110				PRG4	I PRG4 START ADDRESS
1086	001264	012410				PRG5	I PRG5 START ADDRESS
1087	001266	013370				PRG6	I PRG6 START ADDRESS
1088	001270	013454				PRG7	I PRG7 START ADDRESS
1089	001272	013626				PRG10	I PRG10 START ADDRESS
1090	001274	013662				PRG11	I PRG11 START ADDRESS
1091	001276	013672				PRG12	I PRG12 START ADDRESS
1092	001300	013762				PRG13	I PRG13 START ADDRESS
1093	001302	014034				PRG14	I PRG14 START ADDRESS
1094	001304				EMTTAB1		
1095	001304	002666			TYP		I POINTER FOR EMT CALL TYPE
1096	001306	003014			TYP8		I POINTER FOR EMT CALL TYPES
1097	001310	003146			STAL		I POINTER FOR EMT CALL STALL
1098	001312	001466			ERR		I POINTER FOR EMT CALL ERROR
1099	001314	001466			DTCHK		I POINTER FOR EMT CALL DATCHK
1100	001316	001422			CHLT		I POINTER FOR EMT CALL CHALT
1101	001320	002414			STLSRV		I POINTER FOR EMT CALL STDRV
1102	001322	002444			STLSPV		I POINTER FOR EMT CALL STPCHV
1103	001324	001434			EHLT		I POINTER FOR EMT CALL EHLT
1104	001326	002474			SRSETT		I POINTER FOR EMT CALL SRESET
1105	001330	001754			CHAINN		I POINTER FOR EMT CALL SCOPE
1106	001332	002210			CHK33		I POINTER FOR EMT CALL CK33

1107	001334	002226			CHK35		I POINTER FOR EMT CALL CK35
1108	001336	002224			CHK33B		I POINTER FOR EMT CALL CK37
1109	001340	004574			TYPL3		I POINTER FOR EMT CALL TYPLN3
1110	001342	001456			DTHLT		I POINTER FOR EMT CALL DATHLT
1111	001344	002264			SAVRG		I POINTER FOR EMT CALL SAVREG
1112	001346	002324			RSTRG		I POINTER FOR EMT CALL RSTRG
1113	001350	002240			CKASR		I POINTER FOR EMT CALL CKASR
1114	001352	002512			RSETT2		I POINTER FOR EMT CALL RESET2
1115	001354	003046			DLY		I POINTER FOR EMT CALL DELAY
1116	001356	003206			RDLY		I POINTER FOR EMT CALL RDELAY
1117	001360	003306			RSTAL		I POINTER FOR EMT CALL RSTALL
1118	001362	000000			RCNT1	OPEN	I CHARACTER COUNT
1119	001364	000000			CRBUF1	OPEN	I HOLDS ONE CHARACTER FROM READER.
1120	001366	000000			CHR11	OPEN	
1121	001370	000000			CHR21	OPEN	
1122	001372	000000			CHR31	OPEN	
1123	001374	000000			CHR1A1	OPEN	
1124	001376	000000			CHR2A1	OPEN	
1125	001400	000000			CHR3A1	OPEN	
1126	001402	000000			ERCTRI	OPEN	
1127	001404	000000			CTRA1	OPEN	
1128	001406	000000			CTRB1	OPEN	
1129	001410	000000			CTRC1	OPEN	
1130	001412	000000			CTRD1	OPEN	
1131	001414	000000			BRCTRI	OPEN	
1132	001416	000000			DVDND1	OPEN	
1133	001420	000000			DVQUOT1	OPEN	
1134					JCOMMON	HALT ROUTINE	
1135	001422	011600			CHLT1	MOV #16,X0	I DEVELOP ADDRESS OF CALLER.
1136	001424	162700	000002		SUB	#2,X0	
1137	001430	000000			HALT		I HALT, ADDRESS OF CALL INSTRUCTION
1138	001432	000002			RTI		I IN DATA LIGHTS.
1139					JUNCONDITIONAL	ERROR HALT ROUTINE.	
1140	001434	011600			EHLT1	MOV #16,X0	I DEVELOP ADDRESS OF CALLER.
1141	001436	162700	000002		SUB	#2,X0	
1142	001442	000000			HALT		I HALT, ADDR OF ERROR CALL
1143	001444	000002			RTI		I IN DATA LIGHTS.
1144					JDATA CHECK	ROUTINE.	
1145	001446	126767	177712	177711	DTCHK1	CMPS CRBUF,CRBUF+1	I COMPARE EXPECTED AND RECEIVED
1146	001450	001403			BEG	DTCKA	I CHARS, BRANCH IF SAME.
1147	001456	016700	177702		DTHLT1	MOV CRBUF,X0	I MOVE S/B AND WAS CHARS TO R0.
1148	001462	000000			HALT		I DATA ERROR HALT, GOOD CHAR IN
1149							I DATA LIGHTS LEFT, BAD CHAR IN DATA
1150	001464	000002			DTCKA1	RTI	I LIGHTS RIGHT, EXIT.
1151					JCONDITIONAL	ERROR HALT.	
1152	001466	032777	040000	176500	ERR1	BIT #BIT14,0SRPTR	I SCOPE SWITCH SET?
1153	001474	001004			BNE	ERRA	I BR IF YES.
1154	001476	011600			MOV	#16,X0	I DEVELOP CALLER'S ADDRESS.
1155	001500	162700	000002		SUB	#2,X0	
1156	001504	000000			HALT		I ERROR HALT.
1157	001506	000002			RTI		I EXIT.
1158					JROUTINE END	HALT SUBROUTINE.	
1159	001510	032777	100000	176456	SHALT1	BIT #BIT15,0SRPTR	I HALT AT END OF TEST?
1160	001516	001403			BEG	SHLTA	I BR IF NOT.

1161	001520	116700	177514		MOVB	RTNNO,X0	ICURRENT TEST 0 TO R0.
1162	001524	000000			HALT		IROUTINE END HALT.
1163	001526	000207		SHLTAI	RTS	X7	IFEXIT.

1164	001530	012706	001200		START:	MOV	#SPBOT,X6	ISSET BOTTOM OF SP STACK.
1165	001534	005067	176236			CLR	PSW	
1166	001540	012767	000006	176236		MOV	#6,MACHER	
1167	001546	005067	177466			CLR	RTNNO	
1168	001552	005737	000042			TST	#42	ICHAIN OR AUTO-ACCEPTANCE?
1169	001556	001404				BEQ	IS	IBR IF NOT.
1170	001560	004767	001616			JSR	PC,TIMCAL	ICALIBRATE DELAY ROUTINE.
1171	001564	000167	012244			JMP	PRG14	IGO RUN PRG14.
1172	001570	017700	176400		IS:	MOV	#SRPTR,X0	I(#SRPTR) TO R0
1173	001574	042700	177760			BIC	#177760,X0	ILIMIT (SR) TO BITS 3=0
1174	001600	020027	000014			CMF	X0,#14	ICOMPARE (SR) TO PROGRAM LIMIT
1175	001604	101402				CRTA		IVALID PROGRAM NUMBER?
1176	001606	104010			INCPRG:	EHALT		I0, INCORRECT PRG NUMBER
1177	001610	000747				BR	START	ISTART OVER.
1178	001612	005067	177432		CRTA:	CLR	PRGID	
1179	001616	010067	177410			MOV	X0,PRGNUM	ISAVE PROGRAM NUMBER AT PRGNUM
1180	001622	001404				BEQ	CRTB	IBR IF 0.
1181	001624	004767	001552			JSR	PC,TIMCAL	ICALIBRATE DELAY ROUTINE.
1182	001630	016700	177376			MOV	PRGNUM,X0	IPRGNUM BACK TO R0.
1183	001634	000241			CRTB:	CLC		
1184	001636	006100				RDL	X0	IR0X2
1185	001640	000170	001252			JMP	#PRGTAB(0)	IGO TO SELECTED PROGRAM.
1186	001644	104005			SRSET:	CHALT		ISSET SR OPTIONS DESIRED
1187	001646	016767	177362	177366	GETRDY:	MOV	KSTART,NXTST	IADDR OF 1ST ROUTINE TO NXTST
1188	001654	012767	000006	176122	CLEAN:	MOV	#6,MACHER	IRESET MACHER TRAP.
1189	001662	012706	001200			MOV	#SPBOT,X6	ISSET UP BOTTOM OF STACK.
1190	001666	104400				DELAYX		
1191	001670	104011				SRESET		
1192	001672	005067	176100			CLR	PSW	
1193	001676	004767	000210		GTRDYA:	JSR	X7,FORMD	IROLL FORWARD TO "NEXT" ROUTINE.
1194	001702	032777	001000	176264	GTRDYB:	BIT	#BIT0,#SRPTR	ISELECT ROUTINE SWITCH SET?
1195	001710	001002				GTROYC		IBRANCH IF YES.
1196	001712	000177	177320			JMP	#CURTST	IRUN CURRENT ROUTINE.
1197	001716	017700	176252		GTRDYC:	MOV	#SRPTR,X0	I(#SRPTR) TO R0
1198	001722	042700	177600			BIC	#177600,X0	IMASK UNDESIRED BITS
1199	001726	126700	177306			CMFB	RTNNO,X0	ICOMPARE RTNNO TO (R0)
1200	001732	001002				GTRDYD		IBRANCH IF ROUTINE NOT FOUND YET.
1201	001734	000177	177276			JMP	#CURTST	IGO RUN ROUTINE.
1202	001740	022767	177777	177274	GTRDYD:	CMF	0=1,NXTST	I0, CHECK FOR LAST ROUTINE.
1203	001746	001353				SNE	GTRDYA	ILAST ROUTINE?
1204	001750	104010			INCRTN:	EHALT		IYES, INCORRECT ROUTINE SELECTED.
1205	001752	000735				BR	GETRDY	ISTART OVER.
1206	001754	012706	001200		CHAINN:	MOV	#SPBOT,R6	IRESET STACK.
1207	001760	032777	000000	176206		MOV	#BIT14,#SRPTR	ISCOPE?
1208	001766	001406				BEQ	CHNA	IBR IF NOT.
1209	001770	022767	177777	177250		CMF	0=L,SCOPTR	IYES, SCOPE POINTER = -1?
1210	001776	001402				BEQ	CHNA	IBRANCH IF SCOPE ENTRY IS -1.
1211	002000	000177	177242			JMP	#SCOPTR	IRETURN TO ROUTINE.
1212	002004	032777	004000	176162	CHNA:	BIT	#BIT11,#SRPTR	IINHIBIT ITERATION?
1213	002012	001005				SNE	CHNAA	IBR IF YES.
1214	002014	005307	177224			DEC	ICTR	I0, ICTR 0?
1215	002020	001402				BEQ	CHNAA	IBR IF YES.
1216	002022	000177	177220			JMP	#SCOPTR	I0, RETURN TO TEST ROUTINE
1217	002026	004767	177456		CHNAA:	JSR	X7,SHALT	IGO HALT IF HALT SWITCH IS SET

```
1218 002032 032777 001000 176134 CHNB: BIT 0BIT0,0SRPTR ;SELECT ROUTINE?
1219 002040 001302 000000 000000 BNE GETRDY ;BR IF YES.
1220 002042 022767 177172 000000 CMP #=1,NXTST ;NO. LAST TEST?
1221 002050 001301 000000 BNE CLEAN ;BR IF NOT.
1222 002052 013700 000042 CHNC: MOV 0042,R0 ;GET CONTENTS OF 42.
1223 002056 001407 000000 BEQ HERE ;BR IF 0.
1224 002060 000005 000000 RESET
1225 002062 004710 000000 LOGIC: JSR PC,(0) ;RETURN TO MONITOR.
1226 002064 000240 000240 NOP,NOP,NOP ;WORD
1227 002072 000167 011736 JMP PRG14 ;RETURN TO PRG14.
1228 002076 032777 002000 176070 HERE: BIT 0BIT10,0SRPTR ;LOOP PROGRAM?
1229 002104 001260 000000 BNE GETRDY ;BR IF YES.
1230 002106 000000 000000 HALT ;PROGRAM END HALT.
1231 002110 000656 000000 BR GETRDY ;RESTART.
1232 002112 016705 177124 FORWD: MOV NXTST,X5 ;ADDR OF NEXT ROUTINE TO R5.
1233 002116 012567 177116 MOV (5)+,RTNNO ;GET NEXT ROUTINE NUMBER.
1234 002122 012567 177114 MOV (5)+,NXTST ;GET ADDR OF NEXT "NEXT" ROUTINE.
1235 002126 105767 177116 TSTB PRGID ;CHECK IF PROGRAM SCOPE AND I COUNT
1236 002132 100407 000000 FORWDB ;PARAMETERS. BRANCH IF NOT.
1237 002134 012567 177104 MOV (5)+,ICTR ;GET ITERATION COUNT.
1238 002140 012567 177102 MOV (5)+,SCOPTR ;GET SCOPE LOOP ENTRY POINTER.
1239 002144 010567 177066 FORWDA: MOV X5,CURTST ;ADDR OF NOW CURRENT TEST TO CURTST.
1240 002150 000207 000000 RTS X7 ;EXIT FORWD SUBROUTINE.
1241 002152 012767 177777 177066 FORWDB: MOV #=1,SCOPTR ;FORCE "NO SCOPE"
1242 002160 012767 000001 177056 MOV #1,ICTR ;FORCE I COUNT OF 1
1243 002166 000766 000000 BR FORWDA
1244
1245 002170 010046 EMTINT: MOV R0,=(6) ;PUSH R0.
1246 002172 016600 000002 MOV 2(6),R0 ;GET EMT PC.
1247 002176 014000 000000 MOV =(0),R0 ;GET EMT CALL.
1248 002200 006300 000000 ASL R0 ;TIMES 2.
1249 002202 016000 171304 MOV EMTAB=10000(0),R0 ;DEVELOP EMT ADDR.
1250 002206 000200 000000 RTS R0 ;GO TO EMT RTN. RESTORE R0.
1251
1252 002210 032767 000001 177012 ;ROUTINES TO DETERMINE TTY TYPES.
1253 002216 001002 000000 CHK33: BIT #1,TTYTYP ;CHECK FOR 33
1254 002220 062716 000002 BNE CHK33B ;BR IF NOT 33.
1255 002224 000002 000000 CHK33A: ADD #2,(6) ;SET UP 33 EXIT.
1256 002226 032767 000001 176774 CHK33B: RTI ;EXIT.
1257 002234 001371 000000 CHK35: BIT #1,TTYTYP ;CHECK FOR 35.
1258 002236 000002 000000 BNE CHK35A ;BR IF 35.
1259 002240 032767 000010 176762 CKASR: RTI ;NOT 35.
1260 002246 001001 000000 BIT 0BIT3,TTYTYP ;CHECK FOR ASR TTY.
1261 002250 000002 000000 BNE +4 ;BRANCH IF NOT ASR.
1262 002252 022626 000000 RTI ;ASR. EXIT.
1263 002254 012767 000001 176762 POPS2: MOV #1,ICTR ;POP STACK TWICE.
1264 002262 104012 000000 SCOPE ;FORCE I COUNT TO A 1.
1265 ;SCOPE TO BYPASS ROUTINE.
1266 002264 012667 000030 ;SAVE REGS 0 TO 4 SUBROUTINE.
1267 002270 012667 000026 SAVRG: MOV (6)+,SVRPC ;SAVE PC AND PSW.
1268 002274 010446 000000 MOV (6)+,SVRPSW
1269 002276 010346 000000 MOV X4,=(6) ;SAVE REGS 0 = 4
1270 002300 010246 000000 MOV X3,=(6) ;IN STACK.
1271 002302 010146 000000 MOV X2,=(6)
1272 002302 010146 000000 MOV X1,=(6)
```

```
1272 002304 010046 000000 MOV X0,=(6)
1273 002306 016746 000010 MOV SVRPSW,=(6) ;RESTORE PC AND PSW.
1274 002312 016746 000002 MOV SVRPC,=(6)
1275 002316 000002 000000 RTI ;EXIT.
1276 002320 000000 000000 SVRPC: OPEN
1277 002322 000000 000000 SVRPSW: OPEN
1278 ;RESTORE REGS 0 TO 4 SUBROUTINE.
1279 002324 012667 000030 RSTRG: MOV (6)+,RSTPC ;SAVE PC AND PSW.
1280 002330 012667 000026 MOV (6)+,RSTPSW
1281 002334 012600 000000 MOV (6)+,X0 ;RESTORE REGS 0 = 4
1282 002336 012601 000000 MOV (6)+,X1 ;FROM STACK.
1283 002340 012602 000000 MOV (6)+,X2
1284 002342 012603 000000 MOV (6)+,X3
1285 002344 012604 000000 MOV (6)+,X4
1286 002346 016746 000010 MOV RSTPSW,=(6) ;RESTORE PC AND PSW.
1287 002352 016746 000002 MOV RSTPC,=(6)
1288 002356 000002 000000 RTI ;EXIT
1289 002360 000000 000000 RSTPC: OPEN
1290 002362 000000 000000 RSTPSW: OPEN
1291
1292 002364 012767 000310 177022 ;ROUTINE TO FETCH A CHARACTER
1293 002372 005277 176612 AREAD: MOV #200,0RCTR ;SET UP DELAY COUNT.
1294 002376 104400 000000 INC 0TKS ;ENABLE READER.
1295 002400 105777 176604 DELAYX ;WAIT.
1296 002404 100402 000000 TSTB 0TKS ;DONE SET?
1297 002406 104010 000000 BMI ARDB ;BR IF YES.
1298 002410 000765 000000 EHALT ;ERROR. NO RESPONSE FROM READER.
1299 002412 000207 000000 BR AREAD ;TRY AGAIN.
1300 ARDB: RTS X7 ;EXIT
1301 ;ROUTINE TO SET LSR INTERRUPT VECTOR AND PRIORITY
1302 002414 017667 000000 000012 STLSRV: MOV 0(6),STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
1303 002422 062716 000002 ADD #2,*X6 ;SET UP EXIT
1304 002426 016701 176566 MOV TKVTR,X1
1305 002432 012721 000000 STPRA: MOV #OPEN,(1)+ ;SET VECTOR ADDRESS
1306 002436 016721 176560 MOV TKLV, (1)+ ;SET PRIORITY
1307 RTI ;EXIT
1308 ;ROUTINE TO SET LSP INTERRUPT VECTOR AND PRIORITY.
1309 002444 017667 000000 000012 STLSPV: MOV 0(6),STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
1310 002452 062716 000002 ADD #2,*X6 ;SET UP EXIT
1311 002456 016701 176542 STPPA: MOV TPVTR,X1
1312 002462 012721 000000 STPPV: MOV #OPEN,(1)+ ;SET VECTOR ADDRESS.
1313 002466 016721 176534 MOV TPLV, (1)+ ;SET PRIORITY
1314 RTI ;EXIT.
1315 ;ROUTINE TO ISSUE RESET.
1316 002474 012700 052525 SRSETT: MOV #52525,X0 ;DATA TO R0.
1317 002500 005100 000000 COM X0 ;COMPLEMENT (R0).
1318 002502 010067 177770 MOV X0,SRSETT+2 ;(R0) TO SRSETT+2.
1319 002506 000005 000000 RESET ;ISSUE RESET. (R0) IS
1320 002510 000002 000000 RTI ;DISPLAYED. EXIT.
1321 ;DOUBLE RESET SUBROUTINE.
1322 002512 104011 000000 RSETT2: SRESET
1323 002514 104011 000000 SRESET
1324 002516 000002 000000 RTI ;EXIT.
1325 ;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
1326 002520 016700 000042 RNGEN: MOV RP1,X0
```

1326 002524 006100 ROL X0
1327 002526 006100 ROL X0
1328 002530 006700 000034 ADD RP2,X0
1329 002534 010067 000026 MOV X0,RP1
1330 002540 006100 ROL X0
1331 002542 006100 ROL X0
1332 002544 006700 000020 ADD RP2,X0
1333 002550 006100 ROL X0
1334 002552 006100 ROL X0
1335 002554 010067 000010 MOV X0,RP2
1336 002560 016700 000002 MOV RP1,X0
1337 002564 000207 RTS X7 JEXIT, NUMBER IN R0
1338 002566 001233 RP1: 1233
1339 002570 007622 RP2: 7622
1340 002572 104006 BREAD: *STRDY JSET READER VECTOR
1341 002574 002632 BREADB *BREADB ITO BREADB
1342 002576 052777 000101 176404 B18 #101,*TKS IENABLE LSR AND LSRI,
1343 002604 104024 DELAY IAWAIT INTERRUPT,
1344 002606 000310 Z00.
1345 002610 005077 176374 CLR *TKS ICLEAR LSRI ENABLE,
1346 002614 104010 BR EHALLT INO RESPONSE HALT,
1347 002616 000765 BREAD BREAD ITRY AGAIN,
1348 002620 022626 BREADA: POPSP2
1349 002622 117767 176364 176534 MOV *TKB,CRBUF JCHAR READ TO CRBUF,
1350 002630 000207 RTS X7 JEXIT SUBROUTINE,
1351 002632 005077 176352 BREADB: CLR *TKS ICLEAR LSR INTERRUPT ENABLE,
1352 002636 105777 176346 TSTB *TKS ITEST FOR DONE,
1353 002642 100003 BPL BREADC IBRANCH IF DONE NOT SET,
1354 002644 012716 MOV *BREADA,*X6 IMODIFY INTERRUPT EXIT TO BREADA,
1355 002650 000002 RTI IOK, EXIT INTERRUPT,
1356 002652 000000 BREADC: HALT IHALT, DONE BIT NOT SET AFTER INTERRUPT,
1357 002654 012716 002662 MOV *BREADD,(6) IPOINT TO BREADD,
1358 002660 000002 RTI JEXIT INTERRUPT,
1359 002662 022626 BREADD: POPSP2
1360 002664 000742 BR BREAD ITRY AGAIN,
1361 JSUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER,
1362 002666 011600 TYP: MOV *X6,X0 IGET ADDRESS THAT CONTAINS MESSAGE ADDRESS,
1363 002670 062716 000002 ADD #2,*X6 ISET UP EXIT,
1364 002674 011000 MOV *X0,X0 IADDRESS OF MESSAGE TO R0,
1365 002676 112067 000110 TYP: MOVB (0)*,TYPDAT IGET CHARACTER
1366 002702 122767 000100 000102 CMPB #100,TYPDAT ICHECK FOR"CHARACTER
1367 002710 001003 BNE TYPC IBRANCH IF NOT"0",
1368 002712 104024 DELAY IWAIT 100 MSECS,
1369 002714 000144 I00.
1370 002716 000002 RTI ITERMINATOR CHAR, DONE, EXIT,
1371 002720 122767 000045 000064 TYPC: CMPB #45,TYPDAT ICHECK FOR"X",
1372 002726 001416 BEQ TYPF IBRANCH IF"X",
1373 002730 122767 000043 000054 CMPB #43,TYPDAT INOT"X",CHECK FOR"#",
1374 002736 001417 BEQ TYPG IBRANCH IF"#",
1375 002740 004767 000002 JSR X7,TYPD ITYPE CHAR IN TYPDAT
1376 002744 000754 BR TYPB
1377 002746 116777 000040 176242 TYPD: MOVB TYPDAT,*TPB IOUTPUT CHARACTER TO PRINTER
1378 002754 105777 176234 TSTB *TPS IWAIT FOR DONE FLAG,
1379 002760 100375 BPL -=4

1380 002762 000207 RTS X7 JEXIT
1381 002764 112767 000015 000020 TYPF: MOVB #15,TYPDAT IMOVE CARRIAGE RETURN CODE TO TYPDAT
1382 002772 004767 177750 JSR X7,TYPD IGO TYPE CHAR,
1383 002776 112767 000012 000006 TYPG: MOVB #12,TYPDAT IMOVE LF CODE TO TYPDAT,
1384 003004 004767 177736 JSR X7,TYPD IGO TYPE CHAR,
1385 003010 000732 BR TYPB
1386 003012 000000 TYPDAT: OPEN
1387 JSUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
1388 003014 011600 TYPB: MOV *X6,X0 IGET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1389 003016 062716 000002 ADD #2,*X6 IUPDATE TO NEXT MESSAGE ADDRESS
1390 003022 011067 000014 MOV *X0,TYPB IADDRESS OF MESSAGE TO TYPB
1391 003026 022767 177777 000006 CMP #1,TYPB ICHECK FOR TERMINATOR
1392 003034 001001 BNE TYPB IBRANCH IF NOT TERMINATOR,
1393 003036 000002 RTI ITERMINATOR, EXIT
1394 003040 104000 TYPB: TYPB: TYPB ICALL ON TYP SUB TO TYPE MESSAGE
1395 003042 000000 TYPB: OPEN IADDRESS OF MESSAGE GOES HERE
1396 003044 000763 BR TYPB IGO PROCESS NEXT MESSAGE
1397 JSUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
1398 003050 DLCLNT=,*2
1399 003046 011627 000000 DLY: MOV (6),#0 IGET DELAY COUNT ADDRESS,
1400 003052 062716 000002 ADD #2,*X6 ISET UP EXIT ADDRESS
1401 003056 017767 177766 177764 MOV #DLCLNT,DLCLNT IDELAY COUNT TO STACK
1402 003064 005067 174706 CLR PSW ISET PRIORITY 0
1403 003072 MSEC=,*2
1404 003070 012767 000000 000046 DLYA: MOV #0,DLYT I1 MSEC COUNT TO DLYT
1405 003076 016767 000042 000040 DLYB: MOV DLYT,DLYT
1406 003104 016767 000034 000032 MOV DLYT,DLYT
1407 003112 016767 000026 000024 MOV DLYT,DLYT
1408 003120 016767 000020 000016 MOV DLYT,DLYT
1409 003126 005367 000012 DEC DLYT IDECREMENT DLYT,
1410 003132 001361 BNE DLYB IBRANCH IF NOT 0,
1411 003134 005367 177710 DEC DLCLNT IDECREMENT COUNT
1412 003140 001353 BNE DLYA IBR IF NOT DONE DELAYING
1413 003142 000002 RTI JEXIT,
1414 003144 000000 DLYT: OPEN
1415 JSUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS, MAXIMUM STALL
1416 IDETERMINED BY CONTENTS OF LOC STLMSK,
1417 003146 032767 040000 176074 STALL: BIT #BIT14,PRGID ITEST FOR STALLS ALLOWED,
1418 003154 001001 BNE STALAA IALLOWED,
1419 003156 000002 RTI INOT ALLOWED,
1420 003160 004767 177334 STALAA: JSR X7,RNGEN IGO GET RANDOM NUMBER,
1421 003164 046700 000014 BIC STLMSK,X0 I# IN R0, APPLY STALL MASK,
1422 003170 001404 BEQ STALB IBRANCH IF RESULT IS 0,
1423 003172 010067 000002 MOV X0,STALA
1424 003176 104024 DELAY IDELAY
1425 003200 000000 STALB: OPEN IDELAY COUNT
1426 003202 000002 STALB: RTI IDONE, EXIT,
1427 003204 000000 STLMSK: OPEN ISTALL MASK,
1428 IREADER DELAY ROUTINE,
1429 003210 RDLCLNT=,*2
1430 003206 011627 000000 RDLY: MOV (6),#0
1431 003212 062716 000002 ADD #2,(6)
1432 003216 017767 177766 177764 MOV #RDLCLNT,RDLCLNT IGET DELAY COUNT,
1433 003224 005067 174546 CLR PSW

1434 003230 016767 177636 000046 ROLYA1 MOV MSEC,DLYR
1435 003236 016767 000042 000040 ROLYB1 MOV DLYR,DLYR
1436 003244 016767 000034 000032 MOV DLYR,DLYR
1437 003252 016767 000026 000024 MOV DLYR,DLYR
1438 003260 016767 000020 000016 MOV DLYR,DLYR
1439 003266 005367 000012 DEC DLYR
1440 003272 001361 000000 BNE ROLYB]DECREMENT DLYR,
1441 003274 005367 177710 DEC ROLCNT]RBR IF NOT 0.
1442 003300 001353 000000 BNE ROLYA]DECREMENT COUNT,
1443 003302 000000 RTI]RBR IF NOT 0.
1444 003304 000000]EXIT.
1445]READER STALL ROUTINE,
1446 003306 032767 040000 175734 RSTAL1 BIT #BIT14,PRGID]STALL?
1447 003314 001001 BNE RSTLA]RBR IF YES,
1448 003316 000002 RTI]NO, EXIT.
1449 003320 004767 177174 RSTLAI JSR PC,RNGEN]GET RANDOM NUMBER.
1450 003324 046700 177654 BIC STLMSK,X0
1451 003330 001404 BEQ RSTLB
1452 003332 010067 000002 MOV X0,RSTLAA
1453 003336 104025 RDELAY]DELAY.
1454 003340 000000 RSTLAA1 OPEN
1455 003342 000002 RSTLB1 RTI]DONE, EXIT.
1456]SUB TO DELAY X TIME,
1457 003350 DLYR0=DLYX+4
1458 003362 DLYR1=DLYXA+4
1459 003344 012727 000144 000000 DLYX1 MOV #100.,#0]
1460 003352 005067 174420 CLR P8W
1461 003356 012727 001750 000000 DLYXA1 MOV #1000.,#0]COUNT TO DLYR1.
1462 003364 005367 177772 DLYR1 DEC DLYR1]DECREMENT DLYR1.
1463 003370 001375 BNE DLYXB]RBR IF NOT 0.
1464 003372 005367 177752 DEC DLYR0]DECREMENT DLYR0.
1465 003376 001367 BNE DLYXA]RBR IF NOT 0.
1466 003400 000002 RTI]EXIT.
1467]DELAY ROUTINE CALIBRATE ROUTINE.
1468 003402 004767 000050 TIMCAL1 JSR PC,TSPCH]OUTPUT CHAR.
1469 003406 004767 000044 JSR PC,TSPCH]OUTPUT CHAR.
1470 003412 104400 DELAYX
1471 003414 104011 SRESET
1472 003416 005067 175776 CLR DVQUOT
1473 003422 016767 175766 MOV BRCTR,DVDND]SAVE BRCTR CONTENTS.
1474 003430 162767 000144 175760 TIMCLA1 SUB #100.,DVDND]DVDND=100
1475 003436 103403 BLO TIMCLB]RBR IF UNSUCCESSFUL.
1476 003440 005267 175754 INC DVQUOT]INCR QUOTIENT,
1477 003444 000771 BR TIMCLA
1478 003446 016767 175746 177416 TIMCLB1 MOV DVQUOT,MSEC]SAVE 1 MSEC CONSTANT.
1479 003454 000207 RTS PC]EXIT.
1480 003456 104007 TSPCH1 STPCHV]SET UP VECTOR.
1481 003460 003542 TSPCHA
1482 003462 005067 175726 CLR BRCTR
1483 003466 005077 175524 CLR #TPB]LOAD BUFFER WITH 0.
1484 003472 052777 000100 175514 #BIT6,TPS]ENABLE INTERRUPT.
1485 003500 016767 175710 175706 TSPCHC1 MOV BRCTR,BRCTR
1486 003506 016767 175702 175700 MOV BRCTR,BRCTR
1487 003514 016767 175674 175672 MOV BRCTR,BRCTR

1488 003522 016767 175666 175664 MOV BRCTR,BRCTR
1489 003530 005267 175660 INC BRCTR
1490 003534 001361 BNE TSPCHC]RBR IF RESULT NOT 0.
1491 003536 001434 EHLT]NO INTERRUPT FROM PUNCH/PRINTER.
1492 003540 000777 BR]
1493 003542 012716 003550 TSPCHA1 MOV #TSPCHB,(6)]MODIFY INTERRUPT EXIT TP TSPCHB.
1494 003546 000002 RTI]EXIT INTERRUPT.
1495 003550 000207 TSPCHB1 RTS PC]EXIT.
1496]SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
1497 003552 004767 176742 GRCNT1 JSR X7,RNGEN]GET RANDOM NUMBER
1498 003556 046700 000010 BIC RCMSK,X0]APPLY MASK
1499 003562 001773 BEQ GRCNT]TRY AGAIN IF RESULT 0
1500 003564 010067 000004 MOV X0,RNCNT]COUNT TO RNCNT
1501 003570 000207 RTS X7]EXIT.
1502 003572 000000 RCMSK1 OPEN]RANDOM CHARACTER MASK.
1503 003574 000000 RNCNT1 OPEN]RANDOM CHARACTER COUNT.
1504]SUB TO COMPARE READER DATA AGAINST EXPECTED DATA AND REPORT ERRORS.
1505 003576 004767 000262 BCHECK1 JSR X7,GTBIN]GET BIN CHARACTER(IN R0)
1506 003602 110067 175557 MOVB CRBUF,CRBUF+1]S/B CHAR TO CRBUF+1
1507 003606 126767 175552 175551 CMPB CRBUF,CRBUF+1]COMPARE S/B AND WAS CHARS.
1508 003614 001001 BNE .+4]BRANCH IF NOT SAME,
1509 003616 000207 RTS X7]SAME, EXIT.
1510 003620 104017 DATHLT]GO HALT AND DISPLAY DATA.
1511 003622 005367 175554 DEC ERCTR]3 ERRORS?
1512 003626 001002 BNE .+6]BRANCH IF NOT 3 YET,
1513 003630 004767 000002 JSR X7,BSYNC]3 ERRORS, RESYNC READER.
1514 003634 000207 RTS X7]EXIT.
1515]SUBROUTINE TO SYNC THE LSR TO A SPECIAL BINARY COUNT PATTERN TEST TAPE.
1516 003636 004767 000164 BSYNC1 JSR X7,INBIN]INITIALIZE BINARY PATTERN
1517 003642 004767 176724 JSR X7,BREAD]READ CHAR AND STORE AT CHR1
1518 003646 116767 175512 175512 MOVB CRBUF,CHR1
1519 003654 004767 176712 JSR X7,BREAD]READ CHAR AND STORE AT CHR2
1520 003660 116767 175500 175502 MOVB CRBUF,CHR2
1521 003666 004767 176700 JSR X7,BREAD]READ CHAR AND STORE AT CHR3,
1522 003672 116767 175466 175472 MOVB CRBUF,CHR3
1523 003700 004767 000012 JSR X7,SYNCA]GO SYNC
1524 003704 000754 BR BSYNC]NO SYNC, TRY AGAIN,
1525 003706 012767 000003 175466 MOV #3,ERCTR
1526 003714 000207 RTS X7]SUCCESS,EXIT,
1527 003716 012767 001000 000100 SYNCA1 MOV #512.,SYCTRA]512 TO SYCTRA,
1528 003724 004767 000134 000134 JSR X7,GTBIN]BIN CHAR TO CHR1A,
1529 003730 010067 175440 MOV X0,CHR1A
1530 003734 004767 000124 JSR X7,GTBIN]BIN CHAR TO CHR2A,
1531 003740 010067 175432 MOV X0,CHR2A
1532 003744 004767 000114 JSR X7,GTBIN]BIN CHAR TO CHR3A,
1533 003750 010067 175424 MOV X0,CHR3A
1534 003754 026767 175406 175412 CMP CHR1,CHR1A]MATCH?
1535 003762 001013 BNE SYNCC]RBR IF NOT,
1536 003764 026767 175400 175404 CMP CHR2,CHR2A]MATCH?
1537 003772 001007 BNE SYNCC]RBR IF NOT,
1538 003774 026767 175372 175376 CMP CHR3,CHR3A]MATCH?
1539 004002 001003 BNE SYNCC]RBR IF NOT,
1540 004004 062716 000002 ADD #2,(6)]SET UP SYNCED EXIT.
1541 004010 000207 RTS X7]EXIT.

```
1542 004012 005367 000006 SYNCC: DEC SYCTRA )TRIED 512 TIMES?
1543 004016 001342 BNE SYNCB )JBR IF NOT.
1544 004020 104010 EHALT )YES, SYNC ERROR.
1545 004022 000207 RTS X7 )SYNC ERROR EXIT.
1546 004024 000000 SYCTRA: OPEN
1547 )SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS.
1548 004026 012767 177777 000014 INBIN: MOV #=1,RIND )SET ALL VARIABLES
1549 004034 004367 000000 JBR X5,BMOVE )TO MINUS 1.
1550 004040 004050 RIND
1551 004042 004051 RIND+1
1552 004044 000013 11.
1553 004046 000207 RTS X7 )EXIT
1554 004050 000000 RIND: OPEN
1555 004052 000000 PT0: OPEN
1556 004054 000000 PT1: OPEN
1557 004056 000000 PIND: OPEN
1558 004060 000000 PT0P: OPEN
1559 004062 000000 PT1P: OPEN
1560 )SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN R0
1561 004064 016767 177762 177762 GTBIN: MOV PT0,PT1 )PREVIOUS BIN CHAR TO PT1
1562 004072 005167 177756 COM PT1
1563 004076 005167 177746 COM RIND
1564 004102 001002 BNE ,+6
1565 004104 005267 177744 INC PT1
1566 004110 042767 177400 177736 BIC #177400,PT1 )MASK TO 8 BITS
1567 004116 016767 177732 177726 MOV PT1,PT0 )SAVE BIN CHAR IN PT0
1568 004124 016700 177724 MOV PT1,X0 )BIN CHAR TO R0.
1569 004130 000207 RTS X7 )EXIT.
1570 004132 016767 177722 177722 GTBINP: MOV PT0P,PT1P )PREVIOUS BIN CHAR TO PT1P
1571 004140 005167 177716 COM PT1P
1572 004144 005167 177706 COM PIND
1573 004150 001002 BNE ,+6
1574 004152 005267 177704 INC PT1P
1575 004156 042767 177400 177676 BIC #177400,PT1P )MASK TO 8 BITS.
1576 004164 016767 177672 177666 MOV PT1P,PT0P )SAVE BIN CHAR IN PT0P.
1577 004172 016701 177664 MOV PT1P,X1 )BIN CHAR TO R1.
1578 004176 000207 RTS X7 )EXIT.
1579 )OCTAL TO ASCII CONVERT ROUTINES
1580 004200 012500 ACNV6: MOV (5)+,X0 )CONVERT TO 6 ASCII. GET OCTAL ADDRESS
1581 004202 012567 000012 MOV (5)+,ACNV6 )GET ASCII ADDRESS
1582 004206 004767 000052 JSR X7,ACNV )CONVERT TO ASCII
1583 004212 004567 000122 JSR X5,BMOVE )MOVE 6 CHARS TO ASCII ADDRESS
1584 004216 004254 A1ST
1585 004220 000000 ACNV8: OPEN
1586 004222 000006 6
1587 004224 000205 RTS X5 )EXIT
1588 004226 012500 ACNV4: MOV (5)+,X0 )CONVERT TO 4 ASCII. GET OCTAL ADDRESS
1589 004230 012567 000012 MOV (5)+,ACNV4 )GET ASCII ADDRESS
1590 004234 004767 000024 JSR X7,ACNV )CONVERT TO ASCII
1591 004240 004567 000074 JSR X5,BMOVE )MOVE 4 CHARS TO ASCII ADDRESS.
1592 004244 004256 A1ST+2
1593 004246 000000 ACNV2: OPEN
1594 004250 000004 4
1595 004252 000205 RTS X5 )EXIT
```

```
1596 004254 000000 A1ST: OPEN
1597 004256 000000 OPEN
1598 004260 000000 OPEN
1599 004262 000000 ACNVX: OPEN
1600 004264 012701 004262 ACNV: MOV #A1ST+6,X1 )ADDR TO STORE ASCII TO R1
1601 004270 012702 000006 MOV #6,X2 )6 TO R2
1602 004274 011067 177762 MOV #X0,ACNVX )OCTAL WORD TO ACNVX
1603 004300 016703 177756 ACNVH: MOV ACNVX,X3
1604 004304 042703 177770 BIC #177770,X3 )ISOLATE LEAST SIGNIFICANT OCTAL #
1605 004310 062703 000060 ADD #60,X3 )ADD 60 TO CONVERT TO ASCII
1606 004314 110341 MOVX X3,-(1) )STORE ASCII BYTE
1607 004316 006067 177740 ROR ACNVX )MOVE NEXT OCTAL DIGIT TO LEAST
1608 004322 006067 177734 ROR ACNVX )SIGNIFICANT POSITION
1609 004326 006067 177730 ROR ACNVX
1610 004332 005302 DEC X2 )DONE 6 TIMES?
1611 004334 001361 BNE ACNVH )NO, REPEAT.
1612 004336 000207 RTS X7 )YES, EXIT.
1613 )SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1614 004340 104020 BMOVE: SAVREG )SAVE REGS.
1615 004342 012501 MOV (5)+,X1 )GET "FROM" ADDRESS
1616 004344 012502 MOV (5)+,X2 )GET "TO" ADDRESS
1617 004346 012503 MOV (5)+,X3 )GET COUNT
1618 004350 112122 BMOVE: MOVX (1)+,(2)+ )MOVE BYTE
1619 004352 005303 DEC X3 )DECREMENT COUNT
1620 004354 001375 BNE BMOVE )BRANCH IF NOT DONE,
1621 004356 104021 RSTREG )RESTORE REGS.
1622 004360 000205 RTS X5 )DONE EXIT
1623 )SUBROUTINE TO CHECK FOR PUNCH READY.
1624 004362 105777 174626 CPRDY: TSTB #TPS )TEST FOR READY BIT.
1625 004366 100001 BPL CPRDYA )BRANCH IF READY NOT SET.
1626 004370 000207 RTS X7 )OK, EXIT.
1627 004372 104010 CPRDYA: EHALT )NOT READY. HALT.
1628 004374 000772 BR CPRDY
1629 )SUBROUTINE TO PUNCH ON LSP CHARACTER IN REG 0.
1630 004376 004767 177760 LSPCH: JSR X7,CPRDY )GO CHECK FOR PUNCH READY.
1631 004402 010077 174610 MOV X0,PTB )LOAD PUNCH BUFFER.
1632 004406 105777 174602 TSTB #TPS )WAIT FOR DONE.
1633 004412 100375 BPL ,=4
1634 004414 005000 CLR X0
1635 004416 000207 RTS X7 )DONE, EXIT.
1636 )BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
1637 004420 012700 015135 BDCNV: MOV #DECVAL,X0 )SET UP ADDR TO STORE DECIMAL ASCII IN R0
1638 004424 013501 MOV #5,X1 )BINARY VALUE TO R1.
1639 004426 012702 004526 MOV #ADTENP,X2 )ADDR OF TEN POWER STRING TO R2.
1640 004432 012767 000005 MOV #5,CNVCTR )SET UP FOR 5 POWER CONVERSIONS.
1641 004440 012267 000060 BDCNVA: MOV (2)+,TENPWR )MOVE POWER OF TEN VALUE TO TENPWR.
1642 004444 004767 000010 JSR X7,SUBTEN )PERFORM CONVERSION
1643 004450 005367 000044 DEC CNVCTR )DONE 5 CONVERSIONS?
1644 004454 001371 BNE BDCNVA )BRANCH IF NOT YET 5.
1645 004456 000205 RTS X5 )YES, EXIT.
1646 004460 005067 000036 SUBTEN: CLR )CLEAR DIGIT
1647 004464 166701 000034 SUBTNA: SUB TENPWR,X1 )SUBTRACT TEN POWER FROM BINARY VALUE.
1648 004470 103403 BCS )BRANCH IF UNSUCCESSFUL SUBTRACTION.
1649 004472 005267 000024 INC DIGIT
```

1650	004476	000772			BR	SUBTNA			
1651	004500	066701	000020		SUBTNB: ADD	TENPHR,X1			RESTORE SUBTRACTED VALUE.
1652	004504	062767	000060	000010	ADD	#60,DIGIT			CONVERT (DIGIT) TO ASCII
1653	004512	116720	000004		MOV	DIGIT,(0)+			MOVE ASCII CHAR TO DECVAL FIELD.
1654	004516	000207			RTS	X7			EXIT.
1655	004520	000000			CNVCTR: OPEN				
1656	004522	000000			DIGIT: OPEN				
1657	004524	000000			TENPHR: OPEN				
1658	004526	023420			ADTENP: 10000,				
1659	004530	001730			1000.				
1660	004532	000144			10.				
1661	004534	000012			10.				
1662	004536	000001			1				
1663									
1664	004540	012767	000114	000020					
1665	004546	012704	015142						
1666	004552	104002							
1667	004554	112400							
1668	004556	004767	177614						
1669	004562	005367	000004						
1670	004566	001371							
1671	004570	000207							
1672	004572	000000							
1673									
1674	004574	011667	000016						
1675	004600	017767	000012	000010					
1676	004606	062716	000002						
1677	004612	004567	000064						
1678	004616	000000							
1679	004620	042767	040000	174422					
1680	004626	004767	177706						
1681	004632	000002							
1682	004634	112767	000015	010300					
1683	004642	112767	000012	010273					
1684	004650	112767	000015	010376					
1685	004656	112767	000012	010371					
1686	004664	112767	000015	010474					
1687	004672	112767	000012	010467					
1688	004700	000207							
1689									
1690	004702	012567	000004						
1691	004706	004567	177426						
1692	004712	000000							
1693	004714	015144							
1694	004716	000003							
1695	004720	004567	177414						
1696	004724	015144							
1697	004726	015147							
1698	004730	000105							
1699	004732	004567	177402						
1700	004736	015144							
1701	004740	015256							
1702	004742	000110							
1703	004744	000205							

1704									
1705	004746	004567	177366						
1706	004752	014176							
1707	004754	015144							
1708	004756	000077							
1709	004760	004567	177354						
1710	004764	014176							
1711	004766	015243							
1712	004770	000011							
1713	004772	004567	177342						
1714	004776	015144							
1715	005000	015256							
1716	005002	000110							
1717	005004	000207							
1718									
1719	005006	004567	177326						
1720	005012	014162							
1721	005014	015144							
1722	005016	000006							
1723	005020	004567	177314						
1724	005024	015144							
1725	005026	015152							
1726	005030	000102							
1727	005032	004567	177302						
1728	005036	015144							
1729	005040	015256							
1730	005042	000110							
1731	005044	000207							
1732									
1733	005046	004567	177266						
1734	005052	014170							
1735	005054	015144							
1736	005056	000006							
1737	005060	004567	177254						
1738	005064	015144							
1739	005066	015152							
1740	005070	000102							
1741	005072	004567	177242						
1742	005076	015144							
1743	005100	015256							
1744	005102	000110							
1745	005104	000207							
1746									
1747	005106	005777	174100						
1748	005112	105777	174072						
1749	005116	100375							
1750	005120	117767	174866	174236					
1751	005126	116723	174232						
1752	005132	116700	174226						
1753	005136	004767	177234						
1754	005142	000207							

```
.SBTTL PRGO = INPUT-OUTPUT LOGIC TESTS
Z=0
X=1
1755      000000
1756      177777
1757      005144 012767 005156 174062 PRGO1  #P0T0,KSTART  /ADDRESS OF 1ST ROUTINE TO KSTART.
1758      005192 000167 174466          JMP          BRSET  /GO GET STARTED.
1759
1760      *****
1761      P0T0: 0 / PRGO TEST ROUTINE 0 *
1762      #P0T1 /ADDRESS OF NEXT ROUTINE *
1763      1000. /TEST ITERATION COUNT *
1764      P0AA /SCOPE ENTRY POINT *
1765      *****
1766      /TEST ABILITY TO REFERENCE THE KEYBOARD/READER STATUS WORD (TKB)
1767      MOV #P0AE,MACHER /SET UP MACHINE ERROR TRAP.
1768      TST #TKS /REFERENCE CODER STATUS WORD.
1769      SCOPE /SCOPE
1770      P0AE1 ERROR /ERROR, TRAPPED WHEN REFERENCING READER,
1771      SCOPE /STATUS WORD (TKS).
1772      *****
1773      P0T1: 1 / PRGO TEST ROUTINE 1 *
1774      #P0T2 /ADDRESS OF NEXT ROUTINE *
1775      1000. /TEST ITERATION COUNT *
1776      P0BA /SCOPE ENTRY POINT *
1777      *****
1778      /TEST ABILITY TO REFERENCE THE KEYBOARD/READER BUFFER (TKB).
1779      MOV #P0BE,MACHER /SET UP MACHINE ERROR TRAP
1780      TST #TKB /REFERENCE READER BUFFER.
1781      SCOPE /SCOPE
1782      P0BE1 ERROR /ERROR, TRAPPED WHEN REFERENCING
1783      SCOPE /READER BUFFER, (TKB).
1784      *****
1785      P0T2: 2 / PRGO TEST ROUTINE 2 *
1786      #P0T3 /ADDRESS OF NEXT ROUTINE *
1787      1000. /TEST ITERATION COUNT *
1788      P0CA /SCOPE ENTRY POINT *
1789      *****
1790      /TEST ABILITY TO REFERENCE PUNCH/PRINTER STATUS WORD (TPS).
1791      MOV #P0CE,MACHER /SETUP MACHINE ERROR TRAP.
1792      TST #TPS /REFERENCE PUNCH/PRINTER STATUS WORD.
1793      SCOPE /SCOPE
1794      P0CE1 ERROR /ERROR, TRAPPED WHEN REFERENCING
1795      SCOPE /PUNCH/PRINTER STATUS WORD (TPS).
1796      *****
1797      P0T3: 3 / PRGO TEST ROUTINE 3 *
1798      #P0T4 /ADDRESS OF NEXT ROUTINE *
1799      1000. /TEST ITERATION COUNT *
1800      P0DA /SCOPE ENTRY POINT *
1801      *****
1802      /TEST ABILITY TO REFERENCE PUNCH/PRINTER BUFFER (TPB).
1803      MOV #P0DE,MACHER /SETUP MACHINE ERROR TRAP.
1804      TST #TPB /REFERENCE PUNCH/PRINTER BUFFER.
1805      SCOPE /SCOPE
1806      P0DE1 ERROR /ERROR, TRAPPED WHEN REFERENCING
1807      SCOPE /PUNCH/PRINTER BUFFER, (TPS).
1808      *****
```

```
1809      005316 000004 P0T4: 4 / PRGO TEST ROUTINE 4 *
1810      005320 005400 P0T5 /ADDRESS OF NEXT ROUTINE *
1811      005322 001750 P0EA /TEST ITERATION COUNT *
1812      005324 005334 P0EA /SCOPE ENTRY POINT *
1813      *****
1814      /TEST ABILITY TO SET AND CLEAR READER/KYBD IE BIT.
1815      MOV #PRTY7,PSW /SET PRIORITY 7.
1816      BIS #BIT6,#TKS /SET ID BIT IN TKS.
1817      BIT #BIT6,#TKS /CHECK ID BIT IN TKS
1818      BNE P0EB /BRANCH IF ID BIT IS SET.
1819      104003 /ERROR 1 ID BIT NOT SET.
1820      104012 /SCOPE
1821      005356 042777 P0EB: BIC #BIT6,#TKS /CLEAR ID BIT IN TKS
1822      005364 032777 P0EB: BIT #BIT6,#TKS /CHECK ID BIT IN TKS.
1823      005372 001401 BEQ P0EC /BRANCH IF ID BIT IS CLEARED.
1824      005374 104003 /ERROR, ID BIT FAILED TO CLEAR.
1825      005376 104012 /SCOPE
1826      *****
1827      P0T5: 5 / PRGO TEST ROUTINE 5 *
1828      005402 005442 P0T6 /ADDRESS OF NEXT ROUTINE *
1829      005404 000144 P0FA /TEST ITERATION COUNT *
1830      005406 005416 P0FA /SCOPE ENTRY POINT *
1831      *****
1832      /TEST ABILITY TO CLEAR ID BIT WITH RESET INSTRUCTION.
1833      MOV #PRTY7,PSW /SET PRIORITY 7.
1834      BIS #BIT6,#TKS /SET ID BIT IN TKS
1835      SRESET /RESET
1836      BIT #BIT6,#TKS /TEST ID BIT.
1837      BEQ .+4 /BRANCH IF ID BIT IS CLEAR.
1838      005436 104003 /ERROR, RESET FAILED TO CLEAR ID BIT.
1839      005440 104012 /SCOPE
1840      *****
1841      P0T6: 6 / PRGO TEST ROUTINE 6 *
1842      005444 005476 P0T7 /ADDRESS OF NEXT ROUTINE *
1843      005446 000024 P0GA /TEST ITERATION COUNT *
1844      005450 005454 P0GA /SCOPE ENTRY POINT *
1845      *****
1846      /TEST THAT READER DONE BIT SETS SOMETIME AFTER RDR ENABLE.
1847      005452 104022 P0GA: CHKASR
1848      005454 104023 RESETE2
1849      005456 005277 173526 INC #TKS /ENABLE READER.
1850      005462 104000 DELAYX /WAIT.
1851      005464 105777 173520 TSTB #TKS /CHECK FOR DONE
1852      005470 100401 BMI .+4 /BRANCH IF DONE BIT SET.
1853      005472 104003 /DONE NOT SET SOMETIME AFTER RDR ENB.
1854      005474 104012 /SCOPE
1855      *****
1856      P0T7: 7 / PRGO TEST ROUTINE 7 *
1857      005500 005530 P0T10 /ADDRESS OF NEXT ROUTINE *
1858      005502 001750 P0GA /TEST ITERATION COUNT *
1859      005504 005516 P0HA /SCOPE ENTRY POINT *
1860      *****
1861      /TEST THAT DONE BIT READS RELIABLY.
1862      005506 104022 CHKASR
```



```

1863 005510 104023 RESE22
1864 005512 004767 174646 JSR XT,AREAD ;ENABLE READER, COME BACK WHEN DONE SET.
1865 005516 103777 173466 P0HA: TSTB *TKS ;TEST FOR DONE
1866 005522 100401 BHI .+4 ;BRANCH IF DONE FOUND SET.
1867 005524 104003 ERROR ;ERROR, DONE BIT NOT FOUND SET.
1868 005526 104012 SCOPE ;SCOPE
1869
1870 005530 000010 J*****
1871 005532 005564 P0T10: 10 ; PRG0 TEST ROUTINE 10
1872 005534 000024 P0T11 ;ADDRESS OF NEXT ROUTINE
1873 005536 005544 20, ;TEST ITERATION COUNT
1874 P0IA ;SCOPE ENTRY POINT
1875
1876 005540 104022 J*****
1877 005542 104023 ;TEST THAT RESET CLEARS DONE BIT
1878 005544 004767 174614 CHKASR
1879 005550 104011 RESE22
1880 005552 103777 173432 P0IA: JSR XT,AREAD ;ENABLE READER, COME BACK WHEN DONE SET.
1881 005556 100001 SRESE2 ;ISSUE RESET.
1882 005560 104003 TSTB *TKS ;TEST FOR DONE BIT
1883 005562 104012 BPL .+4 ;BRANCH IF DONE BIT RESET.
1884 ERROR ;ERROR, RESET FAILED TO CLEAR DONE.
1885 SCOPE ;SCOPE
1886 005564 000011 J*****
1887 005566 005622 P0T11: 11 ; PRG0 TEST ROUTINE 11
1888 005570 000024 P0T12 ;ADDRESS OF NEXT ROUTINE
1889 005572 005600 20, ;TEST ITERATION COUNT
1890 P0JA ;SCOPE ENTRY POINT
1891
1892 005574 104022 J*****
1893 005576 104023 ;TEST THAT REFERENCING READER DATA BUFFER CLEARS DONE
1894 005600 004767 174560 CHKASR
1895 005604 103777 173402 RESE22
1896 005610 103777 173374 P0JA: JSR XT,AREAD ;ENABLE READER, RETURN WHEN DONE SET.
1897 005614 100001 TSTB *TKS ;REFERENCE READ BUFFER.
1898 005616 104003 BPL .+4 ;TEST FOR DONE BIT
1899 005620 104012 ERROR ;BRANCH IF DONE NOT SET.
1900 SCOPE ;REFERENCE TO BUFFER DID NOT RESET DONE.
1901 ;SCOPE
1902 005622 000012 J*****
1903 005624 005664 P0T12: 12 ; PRG0 TEST ROUTINE 12
1904 005626 000024 P0T13 ;ADDRESS OF NEXT ROUTINE
1905 005630 005634 20, ;TEST ITERATION COUNT
1906 P0KA ;SCOPE ENTRY POINT
1907
1908 005632 104022 J*****
1909 005634 104023 ;CHECK THAT BUSY SETS SOMETIME BEFORE DONE SETS.
1910 005636 005277 173346 P0KA: CHKASR
1911 RESE22
1912 INC *TKS ;ENABLE READER.

```

```

1909 005642 032777 004000 173340 P0KB: BIT #BIT11,*TKS ;BUSY SET?
1910 005650 001004 BNE P0KC ;BR IF YES.
1911 005652 103777 173332 TSTB *TKS ;NO, DONE SET?
1912 005656 100371 BPL P0KB ;BR IF NOT.
1913 005660 104003 ERROR ;BUSY BIT FAILED TO SET BEFORE DONE SET.
1914 005662 104012 P0KC: SCOPE
1915
1916 005664 000013 J*****
1917 005666 005732 P0T13: 13 ; PRG0 TEST ROUTINE 13
1918 005670 000024 P0T14 ;ADDRESS OF NEXT ROUTINE
1919 005672 005676 20, ;TEST ITERATION COUNT
1920 P0LA ;SCOPE ENTRY POINT
1921
1922 005674 104022 J*****
1923 005676 104023 ;TEST THAT DONE IS RESET BY START BIT (WHEN BUSY BECOMES SET).
1924 005700 004767 174460 CHKASR
1925 005704 005277 173300 RESE22
1926 005710 032777 004000 173272 P0LA: JSR XT,AREAD ;ENABLE READER, RETURN WHEN DONE SET.
1927 005716 001774 INC *TKS ;ENABLE READER.
1928 005720 103777 173264 BIT #BIT11,*TKS ;WAIT FOR BUSY TO SET.
1929 005724 100001 BEQ *-6
1930 005726 104003 TSTB *TKS ;TEST FOR DONE BIT.
1931 005730 104012 BPL .+4 ;BRANCH IF DONE NOT SET.
1932 ERROR ;ERROR, START BIT FAILED TO RESET DONE.
1933 SCOPE ;SCOPE
1934 005732 000014 J*****
1935 005734 006006 P0T14: 14 ; PRG0 TEST ROUTINE 14
1936 005736 001750 P0T15 ;ADDRESS OF NEXT ROUTINE
1937 005740 005760 1000, ;TEST ITERATION COUNT
1938 P0MA ;SCOPE ENTRY POINT
1939
1940 005742 104022 J*****
1941 005744 104023 ;TEST THAT READ BUFFER CAN BE READ RELIABLY.
1942 005746 004767 174412 CHKASR
1943 005752 117767 173234 173405 JSR XT,AREAD ;ENABLE READER, RETURN WHEN DONE SET.
1944 005760 117767 173226 173376 P0MA: MOVB *TKB,CBUP+1 ;BUFFER CONTENTS TO CBUP+1
1945 005766 126767 173372 173371 MOVB *TKB,CBUP ;BUFFER CONTENTS TO CBUP
1946 005774 001403 CMPB CBUP,CBUP+1 ;COMPARE CONTENTS OF CBUP AND CBUP+1
1947 005776 016700 173362 BEQ P0MB ;BRANCH IF SAME.
1948 006002 000000 MOV CBUP,*0 ;NOT SAME, ERROR, HALT WITH 1ST READ CHAR
1949 006004 104012 HALT ;IN DATA BYTES LEFT, SUBSEQUENT READ IN DATA BYTES RIGHT
1950 SCOPE
1951 006006 000015 J*****
1952 006010 006056 P0T15: 15 ; PRG0 TEST ROUTINE 15
1953 006012 001750 P0T16 ;ADDRESS OF NEXT ROUTINE
1954 006014 006032 1000, ;TEST ITERATION COUNT
1955 P0NA ;SCOPE ENTRY POINT
1956
1957 006016 104022 J*****
1958 006020 104006 ;TEST THAT READER DONE BIT IS ABLE TO CAUSE INTERRUPT, IF THE INTERRUPT IS
1959 006022 006054 STRDRV ;SERVICED, IT WILL HAVE OCCURRED AT CORRECT VECTOR.
1960 P0NC ;SET UP READER VECTOR TO P0NC
1961 RESE22
1962 006026 004767 174332 P0NA: JSR PC,AREAD ;ENABLE READER, RETURN WHEN DONE SET.
1963 006032 005077 173152 CLR *TKS ;DISABLE READER INTERRUPTS

```

```
1963 006036 005067 171734 CLR PSW /ENABLE READER, RETURN WHEN DONE SET.
1964 006042 052777 000100 173100 B18 #BIT6,#TKS /ENABLE READER INTERRUPT,
1965 006030 000200 NOP
1966 006032 104003 AT20E: ERROR /ERROR, READER FAILED TO INTERRUPT.
1967 006034 104012 P0NC: SCOPE /HERE IF INTERRUPT OCCURS.
1968 /*****
1969 006036 000016 P0T16: 16 / PRGO TEST ROUTINE 16 *
1970 006060 006136 P0T17 P0T17 /ADDRESS OF NEXT ROUTINE *
1971 006062 001730 1000. /TEST ITERATION COUNT *
1972 006064 006102 P0A /SCOPE ENTRY POINT *
1973 /*****
1974 /TEST THAT DONE DOES NOT CAUSE INTERRUPT WITH PROCESSOR AT SAME
1975 /PRIORITY LEVEL AS THE READERS INTERRUPT REQUEST LEVEL.
1976 006066 104022 CHKASR
1977 006070 104006 STRDRV /SET READER VECTOR TO P0OE.
1978 006072 006132 P0OE
1979 006074 104023 RESET2
1980 006076 004767 174202 JSR X7,AREAD /ENABLE READER, RETURN WHEN DONE SET.
1981 006102 005077 173102 P0A1: CLR #TKS /DISABLE READER INTERRUPTS.
1982 006106 016767 173110 171602 MOV TKLVL,PSW /SET PROCESSOR TO SAME PRIORITY AS READER'S.
1983 006114 052777 000100 173066 B18 #BIT6,#TKS /ENABLE READER INTERRUPTS.
1984 006122 000240 NOP /NO OP.
1985 006124 005077 173060 CLR #TKS /OK IF NO INTERRUPT OCCURS.
1986 006130 104012 SCOPE /SCOPE
1987 006132 104003 P0OE: ERROR /ERROR, READER ERRONEOUSLY INTERRUPTED
```

```
1988 /WITH PROCESSOR AT SAME PRIORITY
1989 /LEVEL AS THE READER, OR THE READER
1990 006134 104012 SCOPE /IS AT HIGHER PRIORITY THAN SPECIFIED AT TKLVL.
1991 /*****
1992 006136 000017 P0T17: 17 / PRGO TEST ROUTINE 17 *
1993 006140 006222 P0T20 P0T20 /ADDRESS OF NEXT ROUTINE *
1994 006142 001730 1000. /TEST ITERATION COUNT *
1995 006144 006162 P0A /SCOPE ENTRY POINT *
1996 /*****
1997 /TEST THAT DONE CAUSES INTERRUPT WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
1998 /THAN THE READER'S INTERRUPT PRIORITY LEVEL.
1999 006146 104022 CHKASR
2000 006150 104006 STRDRV /SET READER INTERRUPT SERVICE TO
2001 006152 006214 P0PB /P0PB.
2002 006154 104023 RESET2
2003 006156 004767 174202 JSR X7,AREAD /ENABLE READER, RETURN WHEN DONE SET.
2004 006162 005077 173022 P0PA: CLR #TKS /DISABLE READER INTERRUPTS.
2005 006166 016767 173030 171602 MOV TKLVL,PSW /SET PROCESSOR PRIORITY ONE LEVEL LOWER
2006 006174 102767 000040 171574 SUB #40,PSW /THAN READER.(SPECIFIED AT TKLVL).
2007 006202 052777 000100 173000 B18 #BIT6,#TKS /ENABLE READER INTERRUPTS.
2008 006210 000240 NOP
2009 006212 104003 ERROR /FAILED TO INTERRUPT WITH PC AT PRIORITY ONE LEVEL LOWER
2010 /HERE IF INTERRUPT OCCURS, OK, POP STACK TWICE
2011 006214 005077 172770 P0PB: CLR #TKS /DISABLE READER INTERRUPTS
2012 006220 104012 SCOPE /SCOPE
2013 /*****
2014 006222 000020 P0T20: 20 / PRGO TEST ROUTINE 20 *
2015 006224 005326 P0T21 P0T21 /ADDRESS OF NEXT ROUTINE *
2016 006226 001730 1000. /TEST ITERATION COUNT *
2017 006230 006242 P0A /SCOPE ENTRY POINT *
2018 /*****
2019 /TEST THAT DONE DOES NOT REINTERRUPT AFTER RTI WHEN DONE IS NOT CLEARED.
2020 006232 104022 CHKASR
2021 006234 104023 RESET2
2022 006236 004767 174122 JSR X7,AREAD /ENABLE READER, RETURN WHEN DONE SET.
2023 006242 104006 P0GA: STRDRV /SET READER INTERRUPT SERVICE
2024 006244 006276 P0QC /TO P0QC.
2025 006246 005077 172736 CLR #TKS /DISABLE READER INTERRUPTS.
2026 006252 052777 000100 172730 B18 #BIT6,#TKS /ENABLE READER INTERRUPTS.
2027 006260 005067 171512 CLR PSW /SET PRIORITY 0.
2028 006264 000240 NOP
2029 006266 104003 ERROR /ERROR: FAILED TO INTERRUPT
2030 006270 005077 172714 P0OC: CLR #TKS /DISABLE READER INTERRUPTS.
2031 006274 104012 SCOPE /SCOPE
2032 006276 012777 006316 172714 P0OC: MOV #P0QE,#TKVTR /CHANGE INTERRUPT VECTOR TO P0QE
2033 006304 012716 006312 MOV #P0QD,#% /
2034 006310 000002 RTI /RETURN FROM INTERRUPT
2035 006312 000240 P0QD: NOP
2036 006314 000401 BR /+4
2037 006316 104003 P0QE: ERROR /OK IF NO REINTERRUPT OCCURS.
2038 /DONE REINTERRUPTED AFTER
2039 006320 005077 172664 CLR #TKS /DISABLE READER INTERRUPTS.
2040 006324 104012 SCOPE /SCOPE.
2041 /*****
```

```
2042 006326 000021 P0T21: 21 / PRG0 TEST ROUTINE 21 *
2043 006330 004410 P0T22 / ADDRESS OF NEXT ROUTINE *
2044 006332 001750 1000. / TEST ITERATION COUNT *
2045 006334 006344 P0RA / SCOPE ENTRY POINT *
*****
/TEST ABILITY TO SET AND CLEAR PUNCH ID BIT
2046
2047
2048 006336 012767 000340 171432 P0RA: MOV #PRTY7,PSW /SET PRIORITY 7,
2049 006340 052777 000100 172642 B1S #BIT6,0TPS /SET PUNCH ID BIT,
2050 006352 032777 000100 172634 BIT #BIT6,0TPS /CHECK PUNCH ID BIT,
2051 006360 001002 BNE ,+6 /BRANCH IF PUNCH ID BIT IS SET,
2052 006362 104003 ERROR /ERROR1, PUNCH ID BIT DID NOT SET,
2053 006364 104012 SCOPE
2054 006366 042777 000100 172620 BIC #BIT6,0TPS /CLEAR PUNCH ID BIT,
2055 006374 032777 000100 172612 BIT #BIT6,0TPS /CHECK PUNCH ID BIT,
2056 006402 001401 BEQ /BRANCH IF PUNCH ID BIT IS CLEAR
2057 006404 104003 ERROR /ERROR2, PUNCH ID BIT FAILED TO CLEAR,
2058 006406 104012 SCOPE
*****
/TEST ABILITY TO CLEAR PUNCH ID BIT WITH RESET INSTRUCTION
2059
2060 006410 000022 P0T22: 22 / PRG0 TEST ROUTINE 22 *
2061 006412 006452 P0T23 / ADDRESS OF NEXT ROUTINE *
2062 006414 000024 20. / TEST ITERATION COUNT *
2063 006416 006426 P0SA / SCOPE ENTRY POINT *
*****
/TEST ABILITY TO CLEAR PUNCH ID BIT WITH RESET INSTRUCTION
2064
2065
2066 006420 012767 000340 171350 P0SA: MOV #PRTY7,PSW /SET PRIORITY 7,
2067 006426 052777 000100 172560 B1S #BIT6,0TPS /SET PUNCH ID BIT,
2068 006434 104011 SRESET /RESET
2069 006436 032777 000100 172550 BIT #BIT6,0TPS /CHECK PUNCH ID BIT,
2070 006444 001401 BEQ /BRANCH IF PUNCH ID BIT IS CLEAR,
2071 006446 104003 ERROR /ERROR, RESET FAILED TO CLEAR PUNCH ID BIT,
2072 006450 104012 SCOPE
*****
/TEST ABILITY TO SET AND CLEAR THE PUNCH MAINTENANCE BIT
2073
2074 006452 000023 P0T23: 23 / PRG0 TEST ROUTINE 23 *
2075 006454 006326 P0T24 / ADDRESS OF NEXT ROUTINE *
2076 006456 001750 1000. / TEST ITERATION COUNT *
2077 006460 006462 P0TA / SCOPE ENTRY POINT *
*****
/TEST ABILITY TO SET AND CLEAR THE PUNCH MAINTENANCE BIT
2078
2079
2080 006462 052777 000004 172524 P0TA: B1S #BIT2,0TPS /SET MAINTENANCE BIT,
2081 006470 032777 000004 172516 BIT #BIT2,0TPS /CHECK MAINTENANCE BIT
2082 006476 001002 BNE ,+6 /BRANCH IF MAINTENANCE BIT SET,
2083 006500 104003 ERROR /ERROR1, MAINTENANCE BIT FAILED TO SET,
2084 006502 104012 SCOPE
2085 006504 042777 000004 172502 BIC #BIT2,0TPS /CLEAR MAINTENANCE BIT,
2086 006512 032777 000004 172474 BIT #BIT2,0TPS /CHECK MAINTENANCE BIT
2087 006520 001401 BEQ /BRANCH IF MAINTENANCE BIT IS CLEAR,
2088 006522 104003 ERROR /ERROR2, MAINTENANCE BIT FAILED TO CLEAR,
2089 006524 104012 SCOPE
*****
/TEST ABILITY TO SET AND CLEAR THE PUNCH MAINTENANCE BIT
2090
2091 006526 000024 P0T24: 24 / PRG0 TEST ROUTINE 24 *
2092 006530 006562 P0T25 / ADDRESS OF NEXT ROUTINE *
2093 006532 000024 20. / TEST ITERATION COUNT *
2094 006534 006536 P0UA / SCOPE ENTRY POINT *
*****
```

```
2096 /TEST THAT RESET INSTRUCTION CLEARS THE MAINTENANCE BIT.
2097 006536 052777 000004 172450 P0UA: B1S #BIT2,0TPS /SET MAINTENANCE BIT,
2098 006544 104011 SRESET /ISSUE RESET
2099 006546 032777 000004 172440 BIT #BIT2,0TPS /CHECK MAINTENANCE BIT
2100 006554 001401 BEQ /BRANCH IF MAINTENANCE BIT CLEAR,
2101 006556 104003 ERROR /ERROR, RESET FAILED TO CLEAR
2102 006560 104012 SCOPE /THE MAINTENANCE BIT, SCOPE,
*****
/TEST THAT RESET SETS THE PUNCH READY BIT, AND THAT READY CAN BE READ RELIABLY.
2103
2104 006562 000025 P0T25: 25 / PRG0 TEST ROUTINE 25 *
2105 006564 006604 P0T26 / ADDRESS OF NEXT ROUTINE *
2106 006566 001750 1000. / TEST ITERATION COUNT *
2107 006570 006572 P0VA / SCOPE ENTRY POINT *
*****
/TEST THAT RESET SETS THE PUNCH READY BIT, AND THAT READY CAN BE READ RELIABLY.
2108
2109
2110 006572 105777 172416 P0VA: TSTB 0TPS /CHECK PUNCH READY,
2111 006576 104001 BMI ,+4 /BRANCH IF PUNCH READY IS SET,
2112 006600 104003 ERROR /ERROR, RESET FAILED TO SET READY, OR FAILED TO READ IT
2113 006602 104012 SCOPE /SCOPE
*****
/TEST THAT BYTE LOAD OF PUNCH BUFFER +1 DOES NOT RESET READY.
2114
2115 006604 000026 P0T26: 26 / PRG0 TEST ROUTINE 26 *
2116 006606 006634 P0T27 / ADDRESS OF NEXT ROUTINE *
2117 006610 000024 20. / TEST ITERATION COUNT *
2118 006612 006614 P0WA / SCOPE ENTRY POINT *
*****
/TEST THAT PUNCH READY RESETS BY LOADING PUNCH BUFFER.
2119
2120
2121 006614 104023 P0WA: RESET2
2122 006616 005077 172374 CLR 0TPB /LOAD PUNCH BUFFER
2123 006622 105777 172366 TSTB 0TPS /CHECK PUNCH READY BIT,
2124 006626 100001 BPL /BR IF PUNCH READY IS CLEAR,
2125 006630 104003 ERROR /ERROR, BUFFER LOAD FAILED TO CLEAR READY,
2126 006632 104012 SCOPE /SCOPE
*****
/TEST THAT PUNCH BECOMES READY SOMETIME AFTER BUFFER LOAD,
2127
2128 006634 000027 P0T27: 27 / PRG0 TEST ROUTINE 27 *
2129 006636 006670 P0T30 / ADDRESS OF NEXT ROUTINE *
2130 006640 000024 20. / TEST ITERATION COUNT *
2131 006642 006644 P0XA / SCOPE ENTRY POINT *
*****
/TEST THAT PUNCH BECOMES READY SOMETIME AFTER BUFFER LOAD,
2132
2133
2134 006644 104023 P0XA: RESET2
2135 006646 016700 172344 MOV TPB,X0
2136 006652 005200 INC X0
2137 006654 105010 CLRB 0X0 /BYTE LOAD PUNCH BUFFER+1
2138 006656 105777 172332 TSTB 0TPS /CHECK PUNCH READY BIT
2139 006662 100401 BMI ,+4 /BRANCH IF PUNCH READY STILL SET,
2140 006664 104003 ERROR /ERROR, BYTE LOAD OF PUNCH BUFFER+1
2141 006666 104012 SCOPE /CLEARED READY, SCOPE
*****
/TEST THAT PUNCH BECOMES READY SOMETIME AFTER BUFFER LOAD,
2142
2143 006670 000030 P0T30: 30 / PRG0 TEST ROUTINE 30 *
2144 006672 006722 P0T31 / ADDRESS OF NEXT ROUTINE *
2145 006674 000024 20. / TEST ITERATION COUNT *
2146 006676 006700 P0YA / SCOPE ENTRY POINT *
*****
/TEST THAT PUNCH BECOMES READY SOMETIME AFTER BUFFER LOAD,
2147
2148
2149 006700 104023 P0YA: RESET2
```

```
2150 006702 005077 172310 CLR #TPB , LOAD PUNCH BUFFER,  
2151 006706 104000 DELAYX ,WAIT,  
2152 006710 105777 172300 TSTB ,CHECK PUNCH READY BIT,  
2153 006714 100401 BMI ,*4 ,BRANCH IF PUNCH READY IS SET,  
2154 006716 104003 ERROR ,READY NOT SET SOMETIME AFTER BUFFER LOAD,  
2155 006720 104012 SCOPE ,SCOPE  
2156 *****  
2157 006722 000031 P0T31: 31 , PRG0 TEST ROUTINE 31 *  
2158 006724 006742 P0T32 ,ADDRESS OF NEXT ROUTINE *  
2159 006726 001750 1000, ,TEST ITERATION COUNT *  
2160 006730 006736 P0ZA ,SCOPE ENTRY POINT *  
2161 *****  
2162 ,TEST THAT PUNCH READY BIT CAN CAUSE AN INTERRUPT, IF THE INTERRUPT  
2163 ,IS SERVICED, IT WILL HAVE OCCURRED AT THE CORRECT VECTOR,  
2164 006732 104007 STPCHV ,SET PUNCH INTERRUPT SERVICE  
2165 006734 006740 P0ZB ,TO P0ZB  
2166 006736 005077 172252 P0ZA: CLR #TPS ,DISABLE PUNCH INTERRUPTS  
2167 006742 005067 171030 CLR PSW ,SET PRIORITY 0,  
2168 006746 052777 000100 172240 BIS #BIT6,#TPS ,ENABLE PUNCH INTERRUPTS,  
2169 006754 000240 NOP ,  
2170 006756 104003 ERROR ,PUNCH READY FAILED TO CAUSE  
2171 006760 104012 P0ZB: SCOPE ,INTERRUPT, SCOPE  
2172 *****  
2173 006762 000032 P0T32: 32 , PRG0 TEST ROUTINE 32 *  
2174 006764 007032 P0T33 ,ADDRESS OF NEXT ROUTINE *  
2175 006766 001750 1000, ,TEST ITERATION COUNT *  
2176 006770 006776 P0AAA ,SCOPE ENTRY POINT *  
2177 *****  
2178 ,TEST THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WITH PROCESSOR  
2179 ,AT SAME PRIORITY LEVEL AS THE PUNCH INTERRUPT REQUEST LEVEL,  
2180 006772 104007 STPCHV ,SET PUNCH INTERRUPT SERVICE  
2181 006774 007026 P0AAE ,TO P0AAE,  
2182 006776 016767 172224 170772 P0AAA: MOV TPLVL,PSW ,SET PROCESSOR TO SAME PRIORITY AS PUNCH,  
2183 007004 005077 172204 CLR #TPS ,DISABLE PUNCH INTERRUPTS,  
2184 007010 052777 000100 172176 BIS #BIT6,#TPS ,ENABLE PUNCH INTERRUPTS,  
2185 007016 000240 NOP ,  
2186 007020 005077 172170 P0AAB: CLR #TPS ,OK IF NO INTERRUPT OCCURS,  
2187 007024 104012 SCOPE ,SCOPE  
2188 007026 104003 P0AAE: ERROR ,ERROR, PUNCH INTERRUPTED WITH PROCESSOR  
2189 007030 000773 BR P0AAB ,SET TO SAVE PRIORITY AS THE PUNCH,  
2190 *****  
2191 007032 000033 P0T33: 33 , PRG0 TEST ROUTINE 33 *  
2192 007034 007106 P0T34 ,ADDRESS OF NEXT ROUTINE *  
2193 007036 001750 1000, ,TEST ITERATION COUNT *  
2194 007040 007046 P0BAA ,SCOPE ENTRY POINT *  
2195 *****  
2196 ,TEST THAT THE PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER  
2197 ,THAN THE PUNCH PRIORITY,  
2198 007042 104007 STPCHV ,SET PUNCH INTERRUPT SERVICE  
2199 007044 007100 P0BAC ,TO P0BAC,  
2200 007046 005077 172142 P0BAA: CLR #TPS ,DISABLE PUNCH INTERRUPTS  
2201 007052 016767 172150 170716 MOV TPLVL,PSW ,SET PROCESSOR PRIORITY ONE LEVEL  
2202 007060 162767 000000 170710 SUB #40,PSW ,LOWER THAN PUNCH PRIORITY  
2203 007066 052777 000100 172120 BIS #BIT6,#TPS ,ENABLE PUNCH INTERRUPTS
```

```
2204 007074 000240 NOP ,  
2205 007076 104003 ERROR ,ERROR, PUNCH FAILED TO INTERRUPT,  
2206 007100 005077 172110 P0BAC: CLR #TPS ,THE STOCK TWICE, DISABLE PUNCH INTERRUPT  
2207 007104 104012 SCOPE ,SCOPE  
2208 *****  
2209 007106 000034 P0T34: 34 , PRG0 TEST ROUTINE 34 *  
2210 007110 007202 P0T35 ,ADDRESS OF NEXT ROUTINE *  
2211 007112 001750 1000, ,TEST ITERATION COUNT *  
2212 007114 007116 P0CAA ,SCOPE ENTRY POINT *  
2213 *****  
2214 ,TEST THAT PUNCH READY DOES NOT REINTERRUPT AFTER RTI WHEN READY  
2215 ,BIT HAS NOT BEEN RESET,  
2216 007116 104007 P0CAA: STPCHV ,SET PUNCH INTERRUPT SERVICE TO  
2217 007120 007152 P0CAC ,TO P0CAC,  
2218 007122 005077 172066 CLR #TPS ,DISABLE PUNCH INTERRUPTS  
2219 007126 005067 170644 CLR PSW ,SET PROCESSOR PRIORITY TO 0  
2220 007132 052777 000100 172054 BIS #BIT6,#TPS ,ENABLE PUNCH INTERRUPTS  
2221 007140 000240 NOP ,  
2222 007142 104003 ERROR ,ERROR 1, PUNCH FAILED TO INTERRUPT,  
2223 007144 005077 172044 CLR #TPS ,DISABLE PUNCH INTERRUPT,  
2224 007150 104012 SCOPE ,SCOPE  
2225 007152 012777 007172 172044 P0CAC: MOV #P0CAE,#TPVTR ,HERE IF INTERRUPT OCCURS, CHANGE  
2226 007160 012716 007166 MOV #P0CAD,#X6 ,PUNCH VECTOR TO P0CAE AND EXIT  
2227 007164 000002 RTI ,INTERRUPT  
2228 007166 000240 P0CAD: NOP ,OK IF NO REINTERRUPT OCCURS  
2229 007170 000401 BR P0CAF ,  
2230 007172 104003 P0CAE: ERROR ,ERR 2, PUNCH REINTERRUPTED AFTER  
2231 007174 005077 172014 P0CAF: CLR #TPS ,RTI WITH READY BIT LEFT ON  
2232 007200 104012 SCOPE ,SCOPE  
2233 *****  
2234 007202 000035 P0T35: 35 , PRG0 TEST ROUTINE 35 *  
2235 007204 007260 P0T36 ,ADDRESS OF NEXT ROUTINE *  
2236 007206 001750 1000, ,TEST ITERATION COUNT *  
2237 007210 007216 P0DAA ,SCOPE ENTRY POINT *  
2238 *****  
2239 ,TEST THAT THE PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING  
2240 ,PROCESSOR PRIORITY TO 0,  
2241 007212 104007 STPCHV ,SET PUNCH INTERRUPT  
2242 007214 007252 P0DAC ,TO P0DAC,  
2243 007216 012767 000340 170552 P0DAA: MOV #PRTY7,PSW ,SET PROCESSOR PRIORITY TO 7,  
2244 007224 005077 171764 CLR #TPS ,DISABLE PUNCH INTERRUPTS  
2245 007230 052777 000100 171756 BIS #BIT6,#TPS ,ENABLE PUNCH INTERRUPTS  
2246 007236 005067 170534 CLR PSW ,LOWER PROCESSOR PRIORITY TO 0,  
2247 007242 012767 000340 170526 MOV #PRTY7,PSW ,RAISE PRIORITY TO 7,  
2248 007250 104003 ERROR ,ERROR, PUNCH FAILED TO INTERRUPT  
2249 ,IMMEDIATELY AFTER CP PRIORITY WAS SET TO 0,  
2250 007252 005077 171736 P0DAC: CLR #TPS ,DISABLE PUNCH INTERRUPTS  
2251 007256 104012 SCOPE ,SCOPE  
2252 *****  
2253 007260 000036 P0T36: 36 , PRG0 TEST ROUTINE 36 *  
2254 007262 007342 P0T37 ,ADDRESS OF NEXT ROUTINE *  
2255 007264 000024 20, ,TEST ITERATION COUNT *  
2256 007266 007276 P0EAA ,SCOPE ENTRY POINT *  
2257 *****
```

```

2258                                     )TEST FOR CORRECT OPERATION OF THE WAIT INSTRUCTION. A WAIT INSTRUCTION
2259 )IS PERFORMED WHILE WAITING FOR A PUNCH INTERRUPT. WHEN THE INTERRUPT
2260 )OCCURS, THE SERVICE ROUTINE CHANGES THE WAIT INSTRUCTION TO AN ERROR
2261 )CALL AND THEN EXITS THE INTERRUPT WITH AN RTI. EXITING THE INTERRUPT
2262 )SHOULD RETURN CONTROL TO THE INSTRUCTION FOLLOWING THE WAIT INSTRUCTION.
2263 )IF CONTROL IS INSTEAD RETURNED TO THE SAME LOCATION WHERE THE WAIT
2264 )INSTRUCTION WAS LOCATED AN ERROR CALL WILL OCCUR, INDICATING A FAILURE
2265 )OF THE WAIT INSTRUCTION.
2266 007270 104023          RESET2
2267 007272 104007          STPCHV
2268 007274 007332          P0EAC
2269 007276 012767 000001 000016 P0EAA: MOV #WAIT,P0EAB )SET PUNCH INTERRUPT SERVICE
2270 007304 005077 171706          CLR #TPB )TO P0EAC.
2271 007310 052777 000100 171676          BIS #BIT6,#TPS )MOVE WAIT INSTRUCTION TO P0EAB
2272 007316 005067 170434          CLR PSW )LOAD PUNCH BUFFER (ENABLES PUNCH)
2273 007322 000000          P0EAB: OPEN          BIS          )ENABLE PUNCH INTERRUPTS
2274                                     )SET PRIORITY 0.
2275                                     )THIS LOCATION CAN BE EITHER
2276                                     )A WAIT INSTRUCTION OR AN ERROR CALL.
2277 007324 005077 171664          CLR #TPS          )IF AN ERROR CALL IS EXECUTED, IT
2278 007330 104012          SCOPE          )INDICATES A FAILURE OF THE WAIT INSTRUCTION.
2279 007332 012767 104003 177762 P0EAC: MOV #ERROR,P0EAB )DISABLE PUNCH INTERRUPTS
2280 007340 000002          RTI          )SCOPE
2281                                     )SCOPE ERROR CALL TO P0EAB.
2282 007342 000037          )MOVE WAIT INSTRUCTION TO P0EAB.
2283 007344 007402          )EXIT INTERRUPT.
2284 007346 000024          P0T37: 37          )*****
2285 007350 007354          P0T40          ) PRG0 TEST ROUTINE 37
2286                                     )ADDRESS OF NEXT ROUTINE
2287                                     )TEST ITERATION COUNT
2288                                     )SCOPE ENTRY POINT
2289 007352 104023          )*****
2290 007354 052777 000004 171632 P0FAA: BIS #BIT2,#TPS )SET MAINTENANCE BIT
2291 007362 005077 171630          CLR #TPB )LOAD PUNCH BUFFER
2292 007366 104400          DELAYX          )WAIT
2293 007370 105777 171614          TSTB #TKS )TEST READER DONE BIT
2294 007374 100401          BMI ,+4 )BRANCH IF READER DONE BIT SET.
2295 007376 104003          ERROR          )ERROR, SOMETIME AFTER PUNCH
2296                                     )BUFFER LOAD WITH MAINTENANCE BIT
2297                                     )SET THE READER DONE BIT WAS NOT SET
2298                                     )SCOPE
2299 )*****
2300 007402 000040          P0T40: 40          ) PRG0 TEST ROUTINE 40
2301 007404 007472          P0T41          )ADDRESS OF NEXT ROUTINE
2302 007406 000024          20.          )TEST ITERATION COUNT
2303 007410 007416          P0GAA          )SCOPE ENTRY POINT
2304 )*****
2305 )TEST THAT CLEARING PUNCH READY AND/OR IE BIT CLEARS PUNCH INTERRUPT REQUEST.
2306 007412 104007          STPCHV          )SET PUNCH VECTOR TO P0GAB.
2307 007414 007466          P0GAB: P0GAB
2308 007416 104023          P0GAA: RESET2
2309 007420 012767 000340 170350          MOV #PRTY7,PSW )SET PRIORITY 7.
2310 007426 052777 000100 171560          BIS #BIT6,#TPS )ENABLE PUNCH INTERRUPTS.
2311 007434 005077 171556          CLR #TPB )OUTPUT CHAR.

```

```

2312 007440 105777 171550          TSTB #TPS )WAIT FOR PUNCH READY.
2313 007444 100375          BPL ,+4
2314 007446 005077 171542          CLR #TPS )DISABLE PUNCH INTERRUPTS.
2315 007452 005077 171540          CLR #TPB )LOAD BUFFER TO CLEAR PUNCH READY.
2316 007456 005067 170314          CLR PSW )SET PRIORITY 0.
2317 007462 000240          NOP
2318 007464 104012          SCOPE          )OK IF NO INTERRUPT OCCURS,
2319 007466 104003          P0GAB: ERROR )READY CLEAR AND/OR IE BIT CLEAR DID NOT
2320 007470 104012          SCOPE          )PREVENT PUNCH/PRINTER INTERRUPT.
2321                                     ) (INTERRUPT REQUEST DID NOT CLEAR,)
2322 )*****
2323 007472 000041          P0T41: 41          ) PRG0 TEST ROUTINE 41
2324 007474 007570          P0T42          )ADDRESS OF NEXT ROUTINE
2325 007476 000024          20.          )TEST ITERATION COUNT
2326 007500 007506          P0HAA          )SCOPE ENTRY POINT
2327 )*****
2328 )TEST THAT CLEARING READER DONE AND/OR IE BIT CLEARS READER INTERRUPT REQUEST.
2329 007502 104006          STRDRV          )SET READER VECTOR TO P0HAB,
2330 007504 007564          P0HAB: P0HAB
2331 007506 104023          P0HAA: RESET2
2332 007510 012767 000340 170260          MOV #PRTY7,PSW )SET PRIORITY 7.
2333 007516 052777 000004 171470          BIS #BIT2,#TPS )SET MAINTENANCE MODE.
2334 007524 005077 171466          CLR #TPB )OUTPUT CHAR.
2335 007530 052777 000100 171452          BIS #BIT6,#TKS )ENABLE READER INTERRUPTS.
2336 007536 105777 171446          TSTB #TKS )WAIT FOR READER DONE.
2337 007542 100375          BPL ,+4
2338 007544 005077 171440          CLR #TKS )DISABLE READER INTERRUPTS.
2339 007550 005777 171436          TST #TKB )CLEAR READER DONE.
2340 007554 005067 170216          CLR PSW )SET PRIORITY 0.
2341 007560 000240          NOP
2342 007562 104012          SCOPE          )OK IF NO INTERRUPT OCCURS,
2343 007564 104003          P0HAB: ERROR )DONE CLEARED AND/OR IE CLEARED DID NOT
2344 007566 104012          SCOPE          )PREVENT READER INTERRUPT.
2345 )*****
2346 007570 000042          P0T42: 42          ) PRG0 TEST ROUTINE 42
2347 007572 007620          P0T43          )ADDRESS OF NEXT ROUTINE
2348 007574 001000          1000          )TEST ITERATION COUNT
2349 007576 007600          P0JAA          )SCOPE ENTRY POINT
2350 )*****
2351 )TEST THE DL11A,B KEYBOARD JUMPERS ARE CUT PROPERLY
2352 007600 012777 173476 171402 P0JAA: MOV #173476,#TKS )ATTEMPT TO SET NON-SETTABLE BITS
2353 007606 005777 171376          TST #TKS )DID ANY SET?
2354 007612 001401          BEQ          )NBR IF NO
2355 007614 104003          ERROR          )*****
2356 007616 104012          SCOPE          )*****
2357 )*****
2358 007620 000043          P0T43: 43          ) PRG0 TEST ROUTINE 43
2359 007622 177777          P0TLST          )ADDRESS OF NEXT ROUTINE
2360 007624 001000          1000          )TEST ITERATION COUNT
2361 007626 007630          P0KAA          )SCOPE ENTRY POINT
2362 )*****
2363 )TEST THE DL11A,B PRINTER JUMPERS ARE CUT PROPERLY
2364 007630 012777 177473 171356 P0KAA: MOV #177473,#TPS )ATTEMPT TO SET NON-SETTABLE BITS
2365 007636 022777 000200 171350          CMP #BIT7,#TPS )DID ANY SET?

```

2366	007644	001401		BEG	.,+4		JBR IF NO
2367	007646	104003		ERROR			
2368	007650	104012		SCOPE			

```

2369          ,SBTTL PRG1 READER TEST
2370          Z=1
2371          X=-1
2372 007652 012767 007706 171354 PRG1: MOV #P1T0,KSTART ;SET ADDRESS OF FIRST ROUTINE
2373 007660 012767 177760 173704 MOV #177760,RCMSK
2374 007666 012767 177400 173310 MOV #177400,STLMSK ;SET STALL LIMIT,
2375 007674 052767 040000 171346 BIS #BIT14,PRGID ;ALLOW STALLS
2376 007702 000167 171736 JMP SRSET ;GO GET STARTED.
2377          ;*****
2378 007706 000000 P1T0: 0 ; PRG1 TEST ROUTINE 0 *
2379 007710 007734 P1T1 ;ADDRESS OF NEXT ROUTINE *
2380 007712 003720 2000. ;TEST ITERATION COUNT *
2381 007714 007722 P1AA ;SCOPE ENTRY POINT *
2382          ;*****
2383 ;READ AND CHECK 2000 CHARACTERS OF SPECIAL BINARY COUNT PATTERN, FULL SPEED.
2384 007716 004767 173714 JSR X7,BSYNC ;SYNC READER; SET ERROR COUNTER.
2385 007722 004767 172644 P1AA: JSR X7,BREAD ;GO READ CHARACTER
2386 007726 004767 173644 JSR X7,BCHECK ;GO CHECK CHARACTER READ.
2387 007732 104012 SCOPE ;SCOPE
2388          ;*****
2389 007734 000001 P1T1: 1 ; PRG1 TEST ROUTINE 1 *
2390 007736 007764 P1T2 ;ADDRESS OF NEXT ROUTINE *
2391 007740 001750 1000. ;TEST ITERATION COUNT *
2392 007742 007750 P1BA ;SCOPE ENTRY POINT *
2393          ;*****
2394 ;READ AND CHECK 1000 CHARACTERS OF SPECIAL BINARY COUNT PATTERN,
2395 ;RANDOM STALL BETWEEN CHARACTERS.
2396 007744 004767 173666 P1BA: JSR X7,BSYNC ;SYNC READER; SET ERROR COUNTER
2397 007750 104002 JSR X7,BREAD ;RANDOM STALL
2398 007752 004767 172614 JSR X7,BREAD ;GO READ CHARACTER
2399 007756 004767 173614 JSR X7,BCHECK ;GO CHECK CHARACTER READ
2400 007762 104012 SCOPE ;SCOPE
2401          ;*****
2402 007764 000002 P1T2: 2 ; PRG1 TEST ROUTINE 2 *
2403 007766 177777 P1TST ;ADDRESS OF NEXT ROUTINE *
2404 007770 000310 200. ;TEST ITERATION COUNT *
2405 007772 010000 P1CA ;SCOPE ENTRY POINT *
2406          ;*****
2407 ;READ AND CHECK 200 CHARACTER GROUPS OF SPECIAL BINARY COUNT PATTERN,
2408 ;RANDOM LENGTH
2409 ;GROUPS (BETWEEN 1 AND 15), RANDOM STALL BETWEEN GROUPS (0 TO 127 MSECS).
2410 007774 004767 173636 P1CA: JSR X7,BSYNC ;SYNC READER; SET ERROR COUNTER.
2411 010000 004767 173546 JSR X7,GRCNT ;GENERATE RANDOM CHARACTER COUNT.
2412 010004 104002 JSR X7,BREAD ;RANDOM STALL (0 TO 127 MSECS)
2413 010006 004767 172560 P1CC: JSR X7,BREAD ;GO READ CHARACTER
2414 010012 004767 173560 JSR X7,BCHECK ;GO CHECK CHARACTER READ
2415 010016 005367 173552 DEC RNCNT ;DECREMENT RANDOM CHAR COUNT
2416 010022 001371 BNE P1CC ;GO READ AGAIN IF COUNT NOT 0,
2417 010024 104012 SCOPE ;SCOPE
  
```

```
2418 .8BTTL PRG2=PRINTER TESTS
2419 Z#2
2420 M#-1
2421 MOV #P2T0,K0START ;SET ADDRESS IF 1ST ROUTINE.
2422 B18 #BIT7,PRGID
2423 MOV #177600,STLMSK ;SET STALL LIMIT
2424 JSR X7,S7BFF ;SET UP BUFFER AREA.
2425 JMP BRSET ;GO GET STARTED.
;*****
P2T0: 0 ; PRG2 TEST ROUTINE 0
2426 P2T1 ;ADDRESS OF NEXT ROUTINE
;*****
;CARRIAGE RETURN TEST.
2430 TYPE ;TYPE TITLE.
2431 CRTST
2432 MOV #02,,RCNT
2433 CK37
2434 SUB #9,,RCNT
2435 MOVB #'\\,X0 ;PRINT
2436 JSR X7,LSPCH ;"\"
2437 MOV RCNT,CTRA ;RCNT TO CTRA
2438 DEC CTRA ;DECREMENT CTRA
2439 BNE CT00 ;BRANCH IF NOT 0
2440 ;0. SCOPE
2441 ;SPACE COUNT TO CTRB.
2442 MOV CTRA,CTRB
2443 MOVB #40,X0
2444 JSR X7,LSPCH ;SPACE.
2445 DEC CTRB ;DECREMENT CTRB.
2446 BNE CT0C ;BRANCH IF NOT DONE SPACING.
2447 MOVB #15,X0
2448 JSR X7,LSPCH ;CARRIAGE RETURN.
2449 JSR X7,LSPCH ;DUMMY CYCLE.
2450 MOVB #'/,X0
2451 JSR X7,LSPCH ;PRINT "/".
2452 BR CT0A
;*****
P2T1: 1 ; PRG2 TEST ROUTINE 1
2453 P2T2 ;ADDRESS OF NEXT ROUTINE
;*****
;RIGHT MARGIN TEST
2454 TYPE ;TYPE TITLE
2455 RMTST
2456 MOV #14,,CTRA ;SET UP FOR 33/35
2457 MOVB #RM33B,RMB
2458 CK37
2459 BR CT1A ;37
2460 MOV #15,,CTRA ;NO
2461 MOVB #RM37A,RMB ;YES.
2462 ;SET UP FOR 37.
2463 TYPE ;TYPE-----I
2464 RM33A
2465 DEC CTRA ;DONE N TIMES.
2466 BNE CT1A ;BRANCH IF NOT N TIMES
2467 ;TYPE-I-.
2470 RMB: OPEN
```

```
2472 SCOPE ;SCOPE.
2473 ;*****
2474 P2T2: 2 ; PRG2 TEST ROUTINE 2
2475 P2T3 ;ADDRESS OF NEXT ROUTINE
;*****
;SPACE TEST
2476 TYPE ;TYPE TITLE.
2477 SPTST
2478 MOV #36,,CTRA ;33/35 COUNT TO CTRA.
2479 TYPE ;TYPE SPACE,\.
2480 SPTSTC
2481 DEC CTRA ;DONE TIMES SET IN CTRA?
2482 BNE CT2A ;BRANCH IF NOT DONE
2483 MOV #36,,CTRA ;SET UP CTRA COUNT FOR 33/35
2484 CT2B: MOV #1,CTRB
2485 CT2C: MOV CTRB,CTRC
2486 MOVB #15,X0 ;CARRIAGE RETURN.
2487 JSR X7,LSPCH
2488 JSR X7,LSPCH ;DUMMY CYCLE.
2489 MOVB #40,X0 ;SPACE NUMBER OF TIMES
2490 JSR X7,LSPCH ;SET IN CTRC.
2491 DEC CTRC ;DONE SPACING.
2492 BNE CT2D ;BRANCH IF NOT DONE SPACING.
2493 MOVB #'/,X0 ;DONE, TYPE A "/".
2494 JSR X7,LSPCH
2495 DEC CTRA ;DONE 36 TIMES?
2496 BNE CT2E ;BRANCH IF NOT DONE.
2497 ;DONE, SCOPE.
2498 ADD #2,CTRB ;MODIFY CTRB FOR NEXT TRY.
2499 BR CT2C ;GO DO IT AGAIN.
;*****
P2T3: 3 ; PRG2 TEST ROUTINE 3
2500 P2T4 ;ADDRESS OF NEXT ROUTINE
;*****
;LINE FEED TEST
2501 TYPE ;TYPE TITLE
2502 LFTST
2503 B18 #BIT14,PRGID ;ALLOW STALLS.
2504 MOV #72,,CTRA ;SET 33/35 LINE FEED COUNT.
2505 CK37
2506 BR CT3A ;37?
2507 ADD #9,,CTRA ;INCREMENT LINE FEED COUNT BY 9.
2508 MOVB #'\\,X0 ;TYPE "\"
2509 JSR X7,LSPCH
2510 MOVB #12,X0 ;LINE FEED.
2511 JSR X7,LSPCH
2512 DEC CTRA ;DONE N TIMES?
2513 BNE CT3B ;BRANCH IF NOT DONE.
2514 ;DONE, SCOPE
2515 STALL ;STALL
2516 BR CT3A ;REPEAT
;*****
P2T4: 4 ; PRG2 TEST ROUTINE 4
2517 P2T5 ;ADDRESS OF NEXT ROUTINE
```

```
2526 J*****  
2527 ;TAB TEST  
2528 010524 012767 000011 000074 MOV #9,TCBNT ;SET TAB COUNT.  
2529 010532 104014 CK35 ;39?  
2530 010534 104012 SCOPE INO.  
2531 010536 004567 000040 JBR X9,TPBM ;TYPE MARKERS  
2532 010542 000007 Y  
2533 010544 104000 TYPE  
2534 010546 014315 TBMRK+1  
2535 010550 012767 000007 170626 CT4A: MOV #7,CTRA ;LINE COUNT TO CTRA  
2536 010556 005067 000046 CLR SPCNT ;0 TO SPACE COUNT.  
2537 010562 004767 000044 CT4B: JBR X7,TABP ;GO SPACE=TAB,  
2538 010566 005267 000036 INC SPCNT ;INCREMENT SPACE COUNT.  
2539 010572 005367 170606 DEC CTRA ;DONE 7 LINES?  
2540 010576 001371 BNE CT4B ;BRANCH IF NOT DONE.  
2541 010600 104012 SCOPE ;DONE, SCOPE,  
2542 010602 012567 170576 TPBM: MOV (5)+,CTRA ;TYPE TEST TITLE.  
2543 010606 104000 TYPE  
2544 010610 014300 TBTST  
2545 010612 104000 TPBMA: TYPE ;TYPE MARKERS  
2546 010614 014326 TBMRK1  
2547 010616 005367 170562 DEC CTRA  
2548 010622 001373 BNE TPBMA  
2549 010624 000205 RTS X5 ;EXIT.  
2550 010626 000000 TBCNT: OPEN ;TAB COUNT  
2551 010630 000000 SPCNT: OPEN ;SPACE COUNT  
2552 010632 104000 TABP: TYPE ;CRLF,  
2553 010634 014337 CRLF  
2554 010636 016767 177764 170542 MOV TBCNT,CTRB ;TAB COUNT TO CTRB  
2555 010644 016767 177760 170536 TABPA: MOV SPCNT,CTRC ;SPACE COUNT TO CTRC  
2556 010652 001407 BEQ TABPC ;BRANCH IF SPACE COUNT IS 0.  
2557 010654 112700 000040 TABPB: MOVB #40,X0 ;SPACE  
2558 010660 004767 173512 JBR X7,LSPCH  
2559 010664 005367 170520 DEC CTRC ;DECREMENT SPACE COUNT  
2560 010670 001371 BNE TABPB ;BRANCH IF NOT YET 0.  
2561 010672 112700 000011 TABPC: MOVB #11,X0 ;TAB  
2562 010676 004767 173474 JBR X7,LSPCH  
2563 010702 004767 173470 JBR X7,LSPCH ;DUMMY CYCLE  
2564 010706 004767 173464 JBR X7,LSPCH ;DUMMY CYCLE.  
2565 010712 112700 000057 MOVB #/,X0 ;TYPE "/"  
2566 010716 004767 173454 JBR X7,LSPCH  
2567 010722 005367 170460 DEC CTRB ;DECREMENT TAB COUNT.  
2568 010726 001346 BNE TABPA ;BRANCH IF NOT DONE TABBING.  
2569 010730 000207 RTS X7 ;DONE, EXIT.  
2570 J*****  
2571 010732 000005 P2T5: 5 ; PRG2 TEST ROUTINE 5 *  
2572 010734 010750 P2T6 ;ADDRESS OF NEXT ROUTINE *  
2573 J*****  
2574 ;TYPE LINE OF CHARACTERS ABC  
2575 010736 104000 TYPE ;TYPE "CHARACTER TESTS"  
2576 010740 014507 CHRTST  
2577 010742 104016 TYPLN3 ;TYPE LINE  
2578 010744 014176 A  
2579 010746 104012 SCOPE ;SCOPE
```

```
2580 J*****  
2581 010750 000006 P2T6: 6 ; PRG2 TEST ROUTINE 6 *  
2582 010752 010762 P2T7 ;ADDRESS OF NEXT ROUTINE *  
2583 J*****  
2584 ;TYPE LINE OF CHARACTERS DEF  
2585 010754 104016 TYPLN3 ;TYPE LINE  
2586 010756 014201 D  
2587 010760 104012 SCOPE ;SCOPE  
2588 J*****  
2589 010762 000007 P2T7: 7 ; PRG2 TEST ROUTINE 7 *  
2590 010764 010774 P2T10 ;ADDRESS OF NEXT ROUTINE *  
2591 J*****  
2592 ;TYPE LINE OF CHARACTERS GHI  
2593 010766 104016 TYPLN3 ;TYPE LINE  
2594 010770 014204 G  
2595 010772 104012 SCOPE ;SCOPE  
2596 J*****  
2597 010774 000010 P2T10: 10 ; PRG2 TEST ROUTINE 10 *  
2598 010776 011006 P2T11 ;ADDRESS OF NEXT ROUTINE *  
2599 J*****  
2600 ;TYPE LINE OF CHARACTERS OF JKL  
2601 011000 104016 TYPLN3 ;TYPE LINE  
2602 011002 014207 J  
2603 011004 104012 SCOPE ;SCOPE  
2604 J*****  
2605 011006 000011 P2T11: 11 ; PRG2 TEST ROUTINE 11 *  
2606 011010 011020 P2T12 ;ADDRESS OF NEXT ROUTINE *  
2607 J*****  
2608 ;TYPE LINE OF CHARACTERS MNO  
2609 011012 104016 TYPLN3 ;TYPE LINE  
2610 011014 014212 M  
2611 011016 104012 SCOPE ;SCOPE  
2612 J*****  
2613 011020 000012 P2T12: 12 ; PRG2 TEST ROUTINE 12 *  
2614 011022 011032 P2T13 ;ADDRESS OF NEXT ROUTINE *  
2615 J*****  
2616 ;TYPE LINE OF CHARACTERS POR  
2617 011024 104016 TYPLN3 ;TYPE LINE  
2618 011026 014215 P  
2619 011030 104012 SCOPE ;SCOPE  
2620 J*****  
2621 011032 000013 P2T13: 13 ; PRG2 TEST ROUTINE 13 *  
2622 011034 011044 P2T14 ;ADDRESS OF NEXT ROUTINE *  
2623 J*****  
2624 ;TYPE LINE OF CHARACTERS STU  
2625 011036 104016 TYPLN3 ;TYPE LINE  
2626 011040 014220 S  
2627 011042 104012 SCOPE ;SCOPE  
2628 J*****  
2629 011044 000014 P2T14: 14 ; PRG2 TEST ROUTINE 14 *  
2630 011046 011056 P2T15 ;ADDRESS OF NEXT ROUTINE *  
2631 J*****  
2632 ;TYPE LINE OF CHARACTERS VWX  
2633 011050 104016 TYPLN3 ;TYPE LINE
```


2634	011052	014223	V		
2635	011054	104012	SCOPE	/SCOPE	
2636			*****		
2637	011056	000015	P2T15	15	/ PRG2 TEST ROUTINE 15
2638	011060	011070	P2T16		/ ADDRESS OF NEXT ROUTINE
2639			*****		
2640			/TYPE LINE OF CHARACTERS YZ8		
2641	011062	104016	TYPLN3		/TYPE LINE
2642	011064	014226	Y		
2643	011066	104012	SCOPE	/SCOPE	
2644			*****		
2645	011070	000016	P2T16	16	/ PRG2 TEST ROUTINE 16
2646	011072	011102	P2T17		/ ADDRESS OF NEXT ROUTINE
2647			*****		
2648			/TYPE LINE OF CHARACTERS 123		
2649	011074	104016	TYPLN3		/TYPE LINE
2650	011076	014231	ONE		
2651	011100	104012	SCOPE	/SCOPE	
2652			*****		
2653	011102	000017	P2T17	17	/ PRG2 TEST ROUTINE 17
2654	011104	011114	P2T20		/ ADDRESS OF NEXT ROUTINE
2655			*****		
2656			/TYPE LINE OF CHARACTERS 456		
2657	011106	104016	TYPLN3		/TYPE LINE
2658	011110	014234	FOUR		
2659	011112	104012	SCOPE	/SCOPE	
2660			*****		
2661	011114	000020	P2T20	20	/ PRG2 TEST ROUTINE 20
2662	011116	011126	P2T21		/ ADDRESS OF NEXT ROUTINE
2663			*****		
2664			/TYPE LINE OF CHARACTERS 789		
2665	011120	104016	TYPLN3		/TYPE LINE
2666	011122	014237	SEVEN		
2667	011124	104012	SCOPE	/SCOPE	
2668			*****		
2669	011126	000021	P2T21	21	/ PRG2 TEST ROUTINE 21
2670	011130	011140	P2T22		/ ADDRESS OF NEXT ROUTINE
2671			*****		
2672			/TYPE LINE OF CHARACTERS I"#		
2673	011132	104016	TYPLN3		/TYPE LINE
2674	011134	014242	C41		
2675	011136	104012	SCOPE	/SCOPE	
2676			*****		
2677	011140	000022	P2T22	22	/ PRG2 TEST ROUTINE 22
2678	011142	011152	P2T23		/ ADDRESS OF NEXT ROUTINE
2679			*****		
2680			/TYPE LINE OF CHARACTERS S%2		
2681	011144	104016	TYPLN3		/TYPE LINE
2682	011146	014245	C44		
2683	011150	104012	SCOPE	/SCOPE	
2684			*****		
2685	011152	000023	P2T23	23	/ PRG2 TEST ROUTINE 23
2686	011154	011164	P2T24		/ ADDRESS OF NEXT ROUTINE
2687			*****		

2688			/TYPE LINE OF CHARACTERS ^()		
2689	011156	104016	TYPLN3		/TYPE LINE
2690	011160	014250	C47		
2691	011162	104012	SCOPE	/SCOPE	
2692			*****		
2693	011164	000024	P2T24	24	/ PRG2 TEST ROUTINE 24
2694	011166	011176	P2T25		/ ADDRESS OF NEXT ROUTINE
2695			*****		
2696			/TYPE LINE OF CHARACTERS **,		
2697	011170	104016	TYPLN3		/TYPE LINE
2698	011172	014253	C52		
2699	011174	104012	SCOPE	/SCOPE	
2700			*****		
2701	011176	000025	P2T25	25	/ PRG2 TEST ROUTINE 25
2702	011200	011210	P2T26		/ ADDRESS OF NEXT ROUTINE
2703			*****		
2704			/TYPE LINE OF CHARACTERS *,/		
2705	011202	104016	TYPLN3		/TYPE LINE
2706	011204	014256	C55		
2707	011206	104012	SCOPE	/SCOPE	
2708			*****		
2709	011210	000026	P2T26	26	/ PRG2 TEST ROUTINE 26
2710	011212	011222	P2T27		/ ADDRESS OF NEXT ROUTINE
2711			*****		
2712			/TYPE LINE OF CHARACTERS !/ <		
2713	011214	104016	TYPLN3		/TYPE LINE
2714	011216	014261	C72		
2715	011220	104012	SCOPE	/SCOPE	
2716			*****		
2717	011222	000027	P2T27	27	/ PRG2 TEST ROUTINE 27
2718	011224	011234	P2T30		/ ADDRESS OF NEXT ROUTINE
2719			*****		
2720			/TYPE LINE OF CHARACTERS =>?		
2721	011226	104016	TYPLN3		/TYPE LINE
2722	011230	014264	C75		
2723	011232	104012	SCOPE	/SCOPE	
2724			*****		
2725	011234	000030	P2T30	30	/ PRG2 TEST ROUTINE 30
2726	011236	011246	P2T31		/ ADDRESS OF NEXT ROUTINE
2727			*****		
2728			/TYPE LINE OF CHARACTERS #[\		
2729	011240	104016	TYPLN3		/TYPE LINE
2730	011242	014267	C100		
2731	011244	104012	SCOPE	/SCOPE	
2732			*****		
2733	011246	000031	P2T31	31	/ PRG2 TEST ROUTINE 31
2734	011250	011260	P2T32		/ ADDRESS OF NEXT ROUTINE
2735			*****		
2736			/TYPE LINE OF CHARACTERS]*AND LEFT ARROW		
2737	011252	104016	TYPLN3		/TYPE LINE
2738	011254	014272	C135		
2739	011256	104012	SCOPE	/SCOPE	
2740			*****		
2741	011260	000032	P2T32	32	/ PRG2 TEST ROUTINE 32

```
2742 011262 011316
2743
2744
2745 011264 004767 173456
2746 011270 042767 040000 167752
2747 011276 004767 173236
2748 011302 052767 040000 167740
2749 011310 004767 173224
2750 011314 104012
2751
2752 011316 000033
2753 011320 011400
2754
2755
2756 011322 104013
2757 011324 104012
2758 011326 104000
2759 011330 014533
2760 011332 004767 173450
2761 011336 012767 000006 170040
2762 011344 042767 040000 167676
2763 011352 004767 173162
2764 011356 052767 040000 167664
2765 011364 004767 173150
2766 011370 005367 170010
2767 011374 001363
2768 011376 104012
2769
2770 011400 000034
2771 011402 177777
2772
2773
2774 011404 104014
2775 011406 104012
2776 011410 104000
2777 011412 014533
2778 011414 004767 173426
2779 011420 012767 000006 167756
2780 011426 042767 040000 167614
2781 011434 004767 173100
2782 011440 052767 040000 167602
2783 011446 004767 173066
2784 011452 005367 167726
2785 011456 001363
2786 011460 104012

P2T33
*****
)TYPE 2 LINES OF ALL CHARACTERS, FIRST LINE FULL SPEED, SECOND LINE WITH STALLS.
)FILL BUFFER WITH ALL CHARACTERS.
)CLEAR STALL BIT IN PRGID
)TYPE LINE.
)SET STALL BIT IN PRGID
)TYPE LINE.
)SCOPE.
*****
P2T33: 33
) PRG2 TEST ROUTINE 33
)ADDRESS OF NEXT ROUTINE
)TYPE 12 LINES OF ABR33 WORST CASE PATTERN, ALTERNATE LINES WITH STALLS.
)33?
)NO, BYPASS TEST.
)TYPE "WORST CASE PATTERN TEST"
)PATTERN TO BUFFER.
)SET COUNT TO 6
)CLEAR STALL BIT IN PRGID.
)TYPE LINE
)SET STALL BIT IN PRGID.
)TYPE LINE.
)DONE 6 TIMES?
)BRANCH IF NOT 6 TIMES YET.
)DONE, SCOPE.
*****
P2T34: 34
) PRG2 TEST ROUTINE 34
)ADDRESS OF NEXT ROUTINE
)TYPE 12 LINES OF ABR35 WORST CASE PATTERN, ALTERNATE LINES WITH STALLS.
)35?
)NO, BYPASS TEST.
)TYPE "WORST CASE PATTERN TEST"
)PATTERN TO BUFFER.
)SET COUNT TO 6.
)CLEAR STALL BIT IN PRGID.
)TYPE LINE.
)SET STALL BIT IN PRGID.
)TYPE LINE
)DONE 6 TIMES?
)BRANCH IF NOT 6 TIMES YET.
)DONE, SCOPE.
*****
P2T34A: 34A
) PRG2 TEST ROUTINE 34A
)ADDRESS OF NEXT ROUTINE
)TYPE 12 LINES OF ABR35 WORST CASE PATTERN, ALTERNATE LINES WITH STALLS.
)35A?
)NO, BYPASS TEST.
)TYPE "WORST CASE PATTERN TEST"
)PATTERN TO BUFFER.
)SET COUNT TO 6.
)CLEAR STALL BIT IN PRGID.
)TYPE LINE.
)SET STALL BIT IN PRGID.
)TYPE LINE
)DONE 6 TIMES?
)BRANCH IF NOT 6 TIMES YET.
)DONE, SCOPE.
*****
P2T34A: 34A
) PRG2 TEST ROUTINE 34A
)ADDRESS OF NEXT ROUTINE
)TYPE 12 LINES OF ABR35 WORST CASE PATTERN, ALTERNATE LINES WITH STALLS.
)35A?
)NO, BYPASS TEST.
)TYPE "WORST CASE PATTERN TEST"
)PATTERN TO BUFFER.
)SET COUNT TO 6.
)CLEAR STALL BIT IN PRGID.
)TYPE LINE.
)SET STALL BIT IN PRGID.
)TYPE LINE
)DONE 6 TIMES?
)BRANCH IF NOT 6 TIMES YET.
)DONE, SCOPE.
*****
```

```
2787
2788
2789 000003
2790 177777
2791 011462 012767 011510 167544
2792 011470 052767 040000 167552
2793 011476 012767 177400 171500
2794 011504 000167 170134
2795
2796 011510 000000
2797 011512 011566
2798 011514 000005
2799 011516 011520
2800
2801
2802 011520 012767 001000 167634
2803 011526 004767 000216
2804 011532 004767 172270
2805 011536 004767 172322
2806 011542 004767 172630
2807 011546 005367 167610
2808 011552 001371
2809 011554 004767 000206
2810 011560 004767 000230
2811 011564 104012
2812
2813 011566 000001
2814 011570 011646
2815 011572 000005
2816 011574 011576
2817
2818
2819
2820 011576 012767 001000 167556
2821 011604 004767 000140
2822 011610 004767 172212
2823 011614 004767 172244
2824 011620 004767 172552
2825 011624 104002
2826 011626 005367 167530
2827 011632 001370
2828 011634 004767 000126
2829 011640 004767 000150
2830 011644 104012
2831
2832 011646 000002
2833 011650 177777
2834 011652 000005
2835 011654 011664
2836
2837
2838
2839
2840 011656 012767 177760 171706

)SBTTL PRG3=PUNCH TEST
)PRG3 = PUNCH TEST
)Z=3
)X=1
)PRG3: MOV #P3T0,KSTART )ADDR OF 1ST ROUTINE TO KSTART.
)BIS #BIT14,PRGID )FOLLOW STALLS.
)MOV #177400,STLMSK )SET STALL MASK
)SRSET )GO GET STARTED
*****
)PRG3 TEST ROUTINE 0
)ADDRESS OF NEXT ROUTINE
)TEST ITERATION COUNT
)SCOPE ENTRY POINT
*****
)PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 0 (FULL SPEED)
)SET CHARACTER COUNT TO 512
)GO PUNCH FRONT END.
)INITIALIZE SPECIAL BINARY COUNT
)GET BINARY CHARACTER
)GO PUNCH THE CHARACTER
)DECREMENT CHAR COUNT.
)BRANCH IF COUNT NOT YET 0 YET.
)PUNCH TRAILER.
)CHECK DATA PUNCHED.
)SCOPE
*****
)PRG3 TEST ROUTINE 1
)ADDRESS OF NEXT ROUTINE
)TEST ITERATION COUNT
)SCOPE ENTRY POINT
*****
)PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 1 (RANDOM STALLS AFTER
)PUNCHING EACH CHARACTER.)
)SET CHARACTER COUNT TO 512.
)GO PUNCH FRONT END.
)INITIALIZE SPECIAL BINARY COUNT.
)GET BINARY CHARACTER.
)GO PUNCH THE CHARACTER.
)RANDOM STALL.
)DECREMENT CHAR COUNT.
)BRANCH IF COUNT NOT YET 0.
)PUNCH TRAILER.
)CHECK DATA PUNCHED.
)SCOPE
*****
)PRG3 TEST ROUTINE 2
)ADDRESS OF NEXT ROUTINE
)TEST ITERATION COUNT
)SCOPE ENTRY POINT
*****
)PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 2.
)(RANDOM STALL BEFORE PUNCHING RANDOM LENGTH GROUP OF CHARACTERS).
)MAXIMUM GROUP LENGTH: 151
)MOV #177760,RCHMSK )SET CHAR GROUP MASK FOR 17(8) MAX.
```

2841	011664	012767	001000	167470	P3CA:	MOV	#512,,RCNT		/SET CHARACTER COUNT TO 512.
2842	011672	004767	000052			JSR	X7,PPRNT		/GO PUNCH FRONT END.
2843	011676	004767	172124			JSR	X7,INBIN		/INITIALIZE SPECIAL BINARY COUNT.
2844	011702	004767	171644		P3CB:	JSR	X7,GRCNT		/GENERATE RANDOM CHARACTER COUNT.
2845	011706	104002				STALL			/RANDOM STALL.
2846	011710	004767	172150		P3CC:	JSR	X7,GTBIN		/GET BINARY CHARACTER.
2847	011714	004767	172456			JSR	X7,L3PCH		/PUNCH THE CHARACTER.
2848	011720	005367	167436			DEC	RCNT		/DECREMENT CHAR COUNT.
2849	011724	001404				BEQ	P3CD		/BRANCH IF COUNT IS 0.
2850	011726	005367	171642			DEC	RNCNT		/NOT 0, DECREMENT RANDOM CHAR COUNT.
2851	011732	001366				BNE	P3CC		/BRANCH IF COUNT NOT YET 0.
2852	011734	000762				BR	P3CB		/BRANCH IF COUNT 0.
2853	011736	004767	000024		P3CD:	JSR	X7,PLTLR		/PUNCH TRAILER.
2854	011742	004767	000046			JSR	X7,PCHECK		/CHECK DATA PUNCHED.
2855	011746	104012				SCOPE			/SCOPE.
2856						JROUTINE TO PUNCH FRONT END.			
2857	011750	004767	000012		PPRNT:	JSR	X7,PLTLR		/PUNCH LEADER
2858	011754	012700	000377			MOV	#377,X0		
2859	011760	004767	172412			JSR	X7,L3PCH		/PUNCH SYNC CHARACTER, (RUBOUT)
2860	011764	000207				RTS	X7		/EXIT.
2861	011766	012767	000106	167410	PLTLR:	MOV	#70,,CTRA		/SET CTRA TO 70.
2862	011774	012700	000177		PLTRA:	MOV	#177,X0		
2863	012000	004767	172372			JSR	X7,L3PCH		/PUNCH CODE 177 FOR LEADER/TRAILER
2864	012004	005367	167374			DEC	CTRA		/PUNCHED 70?
2865	012010	001371				BNE	PLTRA		/BRANCH IF NOT YET 70.
2866	012012	000207				RTS	X7		/DONE EXIT.
2867	012014	012767	000226	167362	PCHECK:	MOV	#150,,CTRA		/SET SYNC COUNT TO 150.
2868	012022	004767	170544		PCHKA:	JSR	X7,BREAD		/READ CHARACTER
2869	012026	122767	000377	167330		CMQB	#377,CRBUF		/IS IT SYNC CHARACTER? (377)
2870	012034	001405				PCHKB			/BRANCH IF SYNC CHAR FOUND.
2871	012036	005367	167342			DEC	CTRA		/NOT FOUND, DECREMENT CTRA.
2872	012042	001367				BNE	PCHKA		/BRANCH IF NOT 150 CHARS READ YET.
2873	012044	104010				EMALT			/150 CHARS READ AND NO SYNC, HALT.
2874	012046	000762				BR	PCHECK		/TRY AGAIN.
2875	012050	004767	171752		PCHKB:	JSR	X7,INBIN		/INITIALIZE BINARY COUNT.
2876	012054	012767	001000	167322		MOV	#512,,CTRA		/SET CHARACTER COUNT TO 512.
2877	012062	004767	170504		PCHKC:	JSR	X7,BREAD		/READ CHARACTER.
2878	012066	004767	171772			JSR	X7,GTBIN		/GET BINARY COUNT CHARACTER.
2879	012072	110067	167267			MOVB	X0,CRBUF+1		
2880	012076	104004				DATCHK			/COMPARE CHARACTERS.
2881	012100	005367	167300		PCHKD:	DEC	CTRA		/512 CHARS READ?
2882	012104	001366				BNE	PCHKC		/BRANCH IF NOT 512 CHARS YET.
2883	012106	000207				RTS	X7		/EXIT.

2884								.SBTTL	PRG4=KEYBOARD TEST
2885		000004					Z=4		
2886		177777					X=-1		
2887	012110	012767	012134	167116	PRG4:	MOV	#P4T0,KSTART		
2888	012116	052767	000200	167124		BIS	#BIT7,PRGID		
2889	012124	104000				TYPE			
2890	012126	014567				KMSG1			
2891	012130	000167	167510			JMP	SRSET		
2892									
2893	012134	000000				P4T0:	0		/ PRG4 TEST ROUTINE 0
2894	012136	012240				P4T1			/ ADDRESS OF NEXT ROUTINE
2895									
2896									
2897	012140	012767	000005	167236		MOV	#5,CTRA		
2898	012146	104006			ET0A:	STRDRV			
2899	012150	012204				ET0B			
2900	012152	104000				TYPE			/TYPE "PRESS A KEY WITHIN 10 SECS."
2901	012154	014605				KMSG2			
2902	012156	052777	000100	167024		BIS	#BIT6,#TKS		/ENABLE KYBD INTERRUPT.
2903	012164	005067	165606			CLR	PSW		
2904	012170	104024				DELAY	10000.		/WAIT 10 SECONDS
2905	012172	023420				TYPE			/TYPE "NO KEYBOARD REQUEST."
2906	012174	104000				KMSG6			
2907	012176	015007				EMALT			/HALT.
2908	012200	104010				BR	ET0CA		
2909	012202	000411				BMI	ET0C		
2910	012204	105777	167000		ET0B:	TSTB	#TKS		/TEST FOR DONE BIT ON
2911	012210	100403				BMI	ET0C		/BRANCH IF DONE BIT SET.
2912	012212	104000				TYPE			/DONE BIT NOT SET, TYPE IF FALSE KEY=
2913	012214	015035				KMSG7			/BOARD OR READER INTERRUPT.
2914	012216	104010				EMALT			/HALT
2915	012220	012716	012226		ET0C:	MOV	#ET0CA,#X6		
2916	012224	000002				RTI			/EXIT INTERRUPT.
2917	012226	104011			ET0CA:	SRESET			
2918	012230	005367	167150			DEC	CTRA		/DONE 5 TIMES?
2919	012234	001344				BNE	ET0A		/BRANCH IF NOT DONE.
2920	012236	104012				SCOPE			/SCOPE
2921									
2922	012240	000001				P4T1:	1		/ PRG4 TEST ROUTINE 1
2923	012242	012320				P4T2			/ ADDRESS OF NEXT ROUTINE
2924									
2925									
2926	012244	104000				TYPE			/TYPE TITLE AND INSTRUCTIONS.
2927	012246	014645				KMSG3			
2928	012250	105777	166734		ET1A:	TSTB	#TKS		/WAIT FOR DONE FLAG
2929	012254	100375				BPL	=4		
2930	012256	117767	166730	167100		MOVB	#TKB,CRBUF		/MOVE KYBD CHAR TO CRBUF.
2931	012264	116777	167074	166724		MOVB	CRBUF,#TPB		/ECHO CHAR READ.
2932	012272	105777	166716			TSTB	#TPB		/WAIT FOR PRINTER DONE.
2933	012276	100375				BPL	=4		
2934	012300	042767	000200	167056		BIC	#BIT7,CRBUF		/CLEAR BIT 7 FROM CRBUF.
2935	012306	122767	000177	167050		CMQB	#177,CRBUF		/COMPARE CRBUF TO RUBOUT (177)
2936	012314	001355				BNE	ET1A		/BRANCH IF NOT RUBOUT (177)
2937	012316	104012				SCOPE			/SCOPE

```
2938                                     J*****  
2939 012320 000002                          P4T2: 2          J PRG4 TEST ROUTINE 2  
2940 012322 177777                          P4TLST         J ADDRESS OF NEXT ROUTINE  
2941                                     J*****  
2942 JOURNAL EQUIVALENT TEST, THE OCTAL EQUIVALENT OF ANY CHARACTER KEYED  
2943 JIS PRINTED, RUBOUT ENDS ROUTINE,  
2944 012324 104001                          TYPES  
2945 012326 014747                          KMSG4          JTYPE TITLE AND INSTRUCTIONS,  
2946 012330 014660                          KMSG3A  
2947 012332 177777                          =1  
2948 012334 005067 167024                  CLR      CRBUF  
2949 012340 105777 166644                  ET2A:1  TSTB  #TKB  
2950 012344 100375                          BPL         =-4  
2951 012346 117767 166640 167010          MOVB     #TKB,CRBUF  
2952 012354 004567 171646                  JSR      %S,ACNV4  
2953 012360 001364                          CRBUF  
2954 012362 015001                          OCTEQV  
2955 012364 104000                          TYPE  
2956 012366 014777                          KMSG5  
2957 012370 002767 000200 166766          BIC      #BIT7,CRBUF  
2958 012376 022767 000177 166760          CMP      #177,CRBUF  
2959 012404 001355                          BNE      ET2A  
2960 012406 104012                          SCOPE  
JSCOPE.
```

```
2961                                     .SBTTL PRG5 COMBINED TEST  
2962 012410 104005                          PRG5:1  CHALT  
2963 012412 004767 172210                  JSR      X7,STBF  
2964 012416 052767 040000 166624          BIS      #BIT14,PRGID  
2965 012424 012767 177600 170552          MOV      #177600,STLMSK  
2966 012432 005067 000054                  CLR      PCHCNT  
2967 012436 005067 000046                  CLR      RBUSY  
2968 012442 104006                          STRDRV  
2969 012444 013064                          RZERO  
2970 012446 104007                          STPCHV  
2971 012450 012734                          PCHDAT  
2972 012452 012767 000241 000612          MOV      #241,SEED0  
2973 012460 004767 000574                  JSR      X7,INIT0  
2974 012464 012767 000077 166714          MOV      #63,CTRB  
2975 012472 112377 166520                  MOVB     (3)+,#TPB  
2976 012476 052777 000100 166510          BIS      #BIT6,#TPS  
2977 012504 104400                          DELAY  
2978 012506 000776                          BR        #-2  
2979 012510 000000                          RBUSY:1 OPEN  
2980 012512 000000                          PCHCNT:1 OPEN  
2981 012514 105777 166474                  TSTPCH:1 TSTB #TPS  
2982 012520 100401                          BMI      .+4  
2983 012522 104010                          EHALT  
2984 012524 005267 177762                  INC      PCHCNT  
2985 012530 000207                          RTS      X7  
2986 012532 026727 177754 000024          PCONT:1 CMP      PCHCNT,#20.  
2987 012540 103424                          BLO      PCONTC  
2988 012542 105767 177742                  TSTB     RBUSY  
2989 012546 100406                          BMI      PCONTA  
2990 012550 052767 000200 177732          BIS      #BIT7,RBUSY  
2991 012556 052777 000101 166424          BIS      #101,#TKS  
2992 012564 026727 177722 000050          PCONTA:1 CMP      PCHCNT,#40.  
2993 012572 101402                          BLOS     PCONTB  
2994 012574 005077 166414                  CLR      #TPS  
2995 012600 032777 000400 165366          PCONTB:1 BIT      #BIT8,#SRPTR  
2996 012606 001001                          BNE      PCONTC  
2997 012610 104002                          STALL  
2998 012612 112377 166400                  PCONTC:1 MOVB     (3)+,#TPB  
2999 012616 000002                          RTI  
3000 012620 105777 166364                  TSTRDR:1 TSTB  #TKS  
3001 012624 100401                          BMI      .+4  
3002 012626 104010                          EHALT  
3003 012630 117767 166356 166526          MOVB     #TKB,CRBUF  
3004 012636 005367 177650                  DEC      PCHCNT  
3005 012642 000207                          RTS      X7  
3006 012644 005767 177642                  RCNT:1  TST      PCHCNT  
3007 012650 001006                          BNE      RCNTA  
3008 012652 042767 000200 177630          BIC      #BIT7,RBUSY  
3009 012660 005077 166324                  CLR      #TKS  
3010 012664 000002                          RTI  
3011 012666 026727 177620 000024          RCNTA:1 CMP      PCHCNT,#20.  
3012 012674 101014                          BHI      RCNTC  
3013 012676 032777 000100 166310          BIT      #BIT6,#TPS  
3014 012704 001003                          BNE      RCNTB  
JRCNTB.
```

3015	012706	052777	000100	166300		BIS	#BIT6,PTP8	JENABLE PUNCH INTERRUPTS.
3016	012714	032777	000400	165252	RCONTB:	BIT	#BIT0,#SRPT8	JCHECK FOR FULL SPEED RUN.
3017	012722	001001				BNE	RCNTC	JBRANCH IF FULL SPEED DESIRED.
3018	012724	104026				RSTALL		JGO STALL.
3019	012726	005277	166256		RCONTC:	INC	*TK8	JENABLE READER.
3020	012732	000002				RTI		JEXIT INTERRUPT.
3021	012734	004767	177554		PCHDAT:	JSR	X7,TSTPCH	JCHECK PUNCH.
3022	012740	003367	166440			DEC	CTRA	J74 CHARS OUTPUTTED?
3023	012744	001272				BNE	PCONT	JBRANCH IF NOT.
3024	012746	005367	166434			DEC	CTRB	J63 LINES OUTPUTTED?
3025	012752	001405				BEQ	PCHDTA	JBRANCH IF YES.
3026	012754	005267	000312			INC	SEED0	JND, SETUP FOR NEXT LINE.
3027	012760	004767	000274			JSR	X7,INIT0	JSETUP LINE, 74 TO CTRA
3028	012764	000662			PCHDTA:	BR	PCONT	JCONTINUE.
3029	012766	105067	002152			BLOCK1	CLRB	JFILL PUNCH BUFFER WITH ZEROES.
3030	012772	004567	171342			JSR	X5,BMOVE	
3031	012776	015144				BLOCK1		
3032	013000	015145				BLOCK1+1		
3033	013002	000107				71.		
3034	013004	012703	015142			MOV	#BLOCKA,X3	JPUNCH BUFFER ADDRESS TO R3.
3035	013010	012767	000024	166366		MOV	#20,,CTRA	JSET CHAR COUNT TO 20.
3036	013016	104007				STPCHV		JSET PUNCH SERVICE TO PCHZER.
3037	013020	013024				PCHZER		
3038	013022	000643				BR	PCONT	JCONTINUE.
3039	013024	004767	177464		PCHZER:	JSR	X7,TSTPCH	JCHECK PUNCH.
3040	013030	005367	166350			DEC	CTRA	JALL CHARS OUTPUTTED?
3041	013034	001236				BNE	PCONT	JBRANCH IF NOT.
3042	013036	012767	000241	000226		MOV	#241,SEED0	JYES
3043	013044	004767	000210			JSR	X7,INIT0	JSETUP LINE, 74 TO CTRA
3044	013050	012767	000077	166330		MOV	#63,,CTRB	JSET LINE COUNT TO 63
3045	013056	104007				STPCHV		JSET PUNCH SERVICE TO PCHDAT.
3046	013060	012734				PCHDAT		
3047	013062	000623				BR	PCONT	JCONTINUE.
3048	013064	004767	177530		RZERO:	JSR	X7,TSTRDR	JCHECK READER.
3049	013070	105767	166270			TSTB	CRBUF	JTEST CHARACTER READ.
3050	013074	001663				BEQ	RCONT	JBRANCH IF 0.
3051	013076	004767	000002			JSR	X7,RZERA	JSET UP TO READ DATA.
3052	013102	000415				BR	RDATA	
3053	013104	012767	000241	000204	RZERA:	MOV	#241,SEED1	JSET UP LINE, 74 TO CTRC
3054	013112	004767	000166			JSR	X7,INIT1	
3055	013116	012767	000077	166266		MOV	#63,,CTRD	JSET LINE COUNT TO 63.
3056	013124	104006				STRDRV		JSET READER SERVICE TO RDATA.
3057	013126	013132				RDATA		
3058	013130	000207				RTS	X7	JEXIT
3059	013132	004767	177462		RDATA:	JSR	X7,TSTRDR	JCHECK READER.
3060	013136	112467	166223			MOV	(4)+,CRBUF+1	JMOVE EXPECTED CHAR TO CRBUF+1
3061	013142	104004				DATCHK		JCHECK DATA.
3062	013144	005367	166240			DEC	CTRC	J74 CHARACTERS CHECKED?
3063	013150	001235				BNE	RCONT	JBRANCH IF NOT.
3064	013152	005367	166234			DEC	CTRD	J63 LINES CHECKED?
3065	013156	001405				BEQ	RDATA	JBRANCH IF YES.
3066	013160	005267	000132			INC	SEED1	JND, SETUP NEXT LINE AND
3067	013164	004767	000114			JSR	X7,INIT1	J74 TO CTRC.
3068	013170	000625				BR	RCONT	JCONTINUE.

3069	013172	105067	002060		RDATA:	CLRB	BLOCK2	JFILL READ BUFFER WITH ZEROES.
3070	013176	004567	171136			JSR	X5,BMOVE	
3071	013202	015256				BLOCK2		
3072	013204	015257				BLOCK2+1		
3073	013206	000107				71.		
3074	013210	012704	015254			MOV	#BLOCKB,X4	JREAD BUFFER ADDRESS TO R4
3075	013214	012767	000024	166166		MOV	#20,,CTRC	JSET CHAR COUNT TO 20.
3076	013222	104006				STRDRV		JSET READER SERVICE TO R20ZER
3077	013224	013230				R20ZER		
3078	013226	000606				BR	RCONT	JCONTINUE.
3079	013230	004767	177364		R20ZER:	JSR	X7,TSTRDR	JCHECK READER
3080	013234	112467	166125			MOV	(4)+,CRBUF+1	JMOVE EXPECTED CHAR TO CRBUF+1
3081	013240	104004				DATCHK		JCHECK DATA.
3082	013242	005367	166142			DEC	CTRC	JALL CHARS CHECKED?
3083	013246	001367				BNE	RDATA	JBRANCH IF NOT.
3084	013250	004767	177630			JSR	X7,RZERA	JSET UP TO READ DATA.
3085	013254	000167	177364			JMP	RCONT	
3086	013260	012703	015142		INIT0:	MOV	#BLOCKA,X3	JPUNCH BUFFER ADDRESS TO R3
3087	013264	004567	000040			JSR	X5,DTFL	JFILL PUNCH BUFFER WITH DATA
3088	013270	015144				BLOCK1		JSTARTING WITH CHAR IN SEED0
3089	013272	000000				OPEN		
3090	013274	012767				MOV	#74,,CTRA	
3091	013302	000207	000112	166102		RTS	X7	JEXIT
3092	013304	012704	015250		INIT1:	MOV	#BLOCKB,X4	JREAD BUFFER ADDRESS TO R4
3093	013310	004567	000014			JSR	X5,DTFL	JFILL READ BUFFER WITH DATA
3094	013314	015256				BLOCK2		JSTARTING WITH A CHAR IN SEED1.
3095	013316	000000				OPEN		
3096	013320	012767	000112	166062		MOV	#74,,CTRC	
3097	013326	000207				RTS	X7	JEXIT
3098	013330	012502			DTFL:	MOV	(5)+,X2	JSTARTING ADDRESS TO R2.
3099	013332	012501				MOV	(5)+,X1	JSEED TO R1.
3100	013334	012767	000110	166020		MOV	#72,,RCNT	JCHAR COUNT TO RCNT.
3101	013342	022701	000340		DTFLA:	CMP	#340,X1	J(R1)EQUAL 340?
3102	013346	001002				BNE	DTFLB	JBRANCH IF NOT.
3103	013350	012701	000241			MOV	#241,X1	JEQUAL, RESET TO 241.
3104	013354	110122			DTFLB:	MOV	X1,(2)+	JMOVE CHAR TO BUFFER
3105	013356	005201				INC	X1	JINCREMENT (R1).
3106	013360	005367	165776			DEC	RCNT	JBUFFER FULL?
3107	013364	001366				BNE	DTFLA	JBRANCH IF NOT.
3108	013366	000205				RTS	X5	JYES, EXIT

```

3109          ,SBTTL PRG6, PRG7
3110          IPRG6=READER EXERCISER, SPECIAL BINARY COUNT PATTERN
3111          I$R13=HALT ON ERROR, $R14=0 STALL, $R14=1 FULL SPEED
3112 013370 012767 177600 167606 PRG6: MOV #177600,$TLMSK I$SET STALL LIMIT
3113 013376 012767 177760 178166      MOV #177760,$RMSK I$SET RANDOM CHARACTER LIMIT.
3114 013404 052767 040000 165636      BIS #BIT14,$PRGID I$ALLOW STALLS.
3115 013412 004767 178220      JSR X7,$R0YNC I$SYNC READER.
3116 013416 004767 178130      GTA: JSR X7,$R0CNT I$GENERATE RANDOM CHAR COUNT.
3117 013422 032777 000400 164544      BIT #BIT0,$SRPTR I$CHECK FOR FULL SPEED RUN
3118 013430 001001      BNE GTB I$BRANCH IF FULL SPEED DESIRED.
3119 013432 104002      STALL I$STALL.
3120 013434 004767 167132      GTB: JSR X7,$READ I$READ CHARACTER
3121 013440 004767 178132      JSR X7,$CHECK I$GO CHECK IT
3122 013444 005367 178124      DEC R0CNT I$DECREMENT CHAR COUNT
3123 013450 001371      BNE GTB I$BRANCH IF COUNT NOT 0.
3124 013452 000761      BR GTA I$COUNT 0, START OVER.
3125          IPRG7=PRINTER EXERCISER, KEYBOARD CONTROLLED,
3126          I$TYPES LINES WITH ANY 5 CHARACTERS, STALLS OR FULL SPEED,
3127 013454 004767 171154      PRG7: JSR X7,$TOP I$SET UP BUFFER.
3128 013460 104000      TYPE I$TYPE TITLE
3129          P$TMG1
3130 013464 015063      HTA: BIS #BIT14,$PRGID I$SET STALL BIT IN PRGID.
3131 013472 012767 177600 167504      MOV #177600,$TLMSK I$SET STALL MASK.
3132 013500 012703 015144      MOV #BLOCK1,$S I$TYPE "TYPE IN DATA".
3133 013504 104000      TYPE
3134 013506 015111      P$TMG2
3135 013510 005777 165476      TST #0KB I$CLEAR BUFFER.
3136 013514 012767 000006 165662      MOV #6,$CTRA I$CHAR COUNT TO CTRA.
3137 013522 004767 171360      HTB: JSR X7,$GKBSCR I$GET AND STORE KYBD CHARACTER.
3138 013526 005367 165652      DEC CTRA I$GOT 6 CHARACTERS?
3139 013532 001373      BNE HTB I$BRANCH IF NOT 6 CHARS YET.
3140 013534 042767 000200 165622      BIC #BIT7,$CRBUF
3141 013542 122767 000177 165614      CMPB #177,$CRBUF I$CHECK 6TH CHAR FOR RUBOUT.
3142 013550 001013      BNE HTC I$BRANCH IF NOT A RUBOUT.
3143 013552 042767 040000 165470      BIC #BIT14,$PRGID I$RUBOUT, CLEAR STALL BIT IN PRGID.
3144 013560 104015      CK37
3145 013562 000406      BR HTC
3146 013564 004567 170550      JSR X5,$BMOVE I$YES, FILL 81 CHAR LINE.
3147 013570 015144      BLOCK1
3148 013572 015151      BLOCK1+5
3149 013574 000114      76.
3150 013576 000405      BR HTD
3151 013600 004567 170534      HTC: JSR X5,$BMOVE I$FILL 72 CHAR LINE.
3152 013604 015144      BLOCK1
3153 013606 015151      BLOCK1+5
3154 013610 000103      67.
3155 013612 004767 170722      HTD: JSR X7,$TYPLN I$TYPE LINE.
3156 013616 005777 164352      TST #SRPTR I$CHANGE DATA? ($R15=1).
3157 013622 100720      BMI HTA I$YES, GO CHANGE DATA
3158 013624 000772      BR HTD I$NO CONTINUE WITH SAME DATA.

```

```

3159          ,SBTTL PRG10, PRG11, PRG12
3160          IPRG10. PUNCH SPECIAL BINARY COUNT PATTERN TEST TAPE
3161 013626 012746 000024      PRG10: MOV #20,$-(6) I$PUNCH 20 BLANK CHAR, LEADER
3162 013632 005000      CLR X0
3163 013634 004767 170536      PRG10A: JSR X7,$LSPCH
3164 013640 005316      DEC #6
3165 013642 001374      BNE PRG10A
3166 013644 004767 170156      JSR X7,$INBIN I$INITIALIZE SPECIAL BINARY COUNT
3167 013650 004767 170210      PRG10B: JSR X7,$GTBIN I$GET BINARY CHARACTER.
3168 013654 004767 170516      JSR X7,$LSPCH I$PUNCH CHARACTER
3169 013660 000773      BR PRG10B I$REPEAT.
3170          ]
3171          ]
3172          IPRG11=PUNCH CLOCK ADJUSTMENT ROUTINE.
3173          I$OUTPUTS CHARACTER SET IN LEFT HALF OF $R, AND
3174          I$STALLS FOR NUMBER OF MILLISECDS SET IN RIGHT HALF OF $R.
3175 013662 104005      PRG11: CHALT I$HALT TO SET $R.
3176 013664 004767 000036      ITA: JSR X7,$C1112 I$GO OUTPUT CHARACTER SET IN LEFT
3177 013670 000775      BR ITA I$HALF OF $R AND STALL PER $R RIGHT.
3178          ]
3179          ]
3180          IPRG12=READER CLOCK ADJUSTMENT ROUTINE.
3181          I$PERFORMS SAME FUNCTION AS PRG11, AND IN ADDITION,
3182          I$USING THE PUNCH MAINTENANCE BIT, SHIFTS OUTPUT OF PUNCH
3183          I$SHIFT REGISTER ONTO THE READER BUFFER, THE CONTENTS OF THE
3184          I$READER BUFFER ARE THEN "FIXED" ON THE CONSOLE DATA LIGHTS
3185          I$BY ISSUING A RESET WITH CONTENTS OF READER BUFFER LOADED IN $R0.
3186 013672 104005      PRG12: CHALT I$HALT TO SET $R.
3187 013674 004767 000020      JTA: JSR X7,$C1112M I$GO OUTPUT CHARACTER FROM $R LEFT AND
3188 013700 017700 165306      MOV #0KB,$X0 I$STALL PER $R RIGHT, (TKB) TO $R0.
3189 013704 000005      RESET I$"FIX" (TKB) IN DATA LIGHTS.
3190 013706 000005      RESET
3191 013710 000005      RESET
3192 013712 000005      RESET
3193 013714 000005      RESET
3194 013716 000766      BR JTA I$REPEAT.
3195          ]
3196 013720 052777 000004 165266      C1112M: BIS #4,$TPS I$SET MAINTENANCE MODE (PUNCH).
3197 013726 117767 164242 000022      C1112: MOVB #SRPTR,$XTY I$STALL COUNT TO $XTY.
3198 013734 005767 000016      TST $XTY I$DISREGARD 0 DELAY.
3199 013740 001002      BNE C1112A
3200 013742 005267 000010      INC $XTY
3201 013746 117777 164223 163242      C1112A: MOVB #SRPTR+1,$TPB I$LOAD PUNCH BUFFER.
3202 013754 104024      DELAY I$DELAY (APPROXIMATELY) THE NUMBER OF
3203 013756 000000      $XTY: OPEN I$MSECS, SPECIFIED AT $R RIGHT
3204 013760 000207      RTS X7 I$EXIT

```

3205
 3206
 3207
 3208
 3209

.0BTTL PRG13, PRG14
 ;PRG13=MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
 ;WITH MAINTENANCE MODE SET, OUTPUTS ONTO PUNCH BUFFER AND BACK ONTO
 ;READER BUFFER THE CHARACTER SET IN SR LEFT. THE CHARACTER IN THE
 ;READER BUFFER IS COMPARED TO THE CHARACTER IN SR LEFT. IF THE 2 CHARACTERS

3210
 3211
 3212
 3213
 3214
 3215
 3216
 3217
 3218
 3219
 3220
 3221
 3222
 3223
 3224
 3225
 3226
 3227
 3228
 3229
 3230
 3231
 3232
 3233
 3234
 3235
 3236
 3237
 3238
 3239
 3240
 3241
 3242
 3243
 3244
 3245
 3246
 3247
 3248
 3249
 3250

;DISAGREE THE PROGRAM HALTS. THE DATA LIGHTS WILL THEN CONTAIN:
 ;
 ;LEFT HALF: THE EXPECTED CHARACTER (SR LEFT).
 ;RIGHT HALF: THE CHARACTER IN THE READER BUFFER.
 PRG13: CHALT ;HALT TO SET SR.
 KTA: B1S #4,PTPS ;SET MAINTENANCE MODE.
 KTB: TSTB #TPS ;WAIT FOR READY.
 BPL ;=4
 MOVB #SRPTR+1,CRBUF+1 ;S/B CHAR TO CRBUF+1.
 MOVB CRBUF+1,PTPB ;OUTPUT CHARACTER.
 TSTB #TKS ;WAIT FOR READER DONE FLAG.
 BPL ;=4
 MOVB #TKB,CRBUF ;CHAR READ TO CRBUF.
 DATCHK ;GO CHECK AGAINST S/B CHAR.
 BR KTA ;REPEAT.
 ;PRG14=MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST.
 ;PERFORMS SAME OPERATION AS PRG13, EXCEPT THAT SPECIAL BINARY COUNT
 ;PATTERN IS USED.
 PRG14: MOV #1024,,CTRA ;SET UP FOR 1024 CHECKS.
 JSR X7,INBIN ;INITIALIZE BINARY COUNT
 MOV #177600,STLMSK ;SET STALL LIMIT
 B1S #BIT14,PRGID ;ALLOW STALLS
 LTA: B1S #4,PTPS ;SET MAINTENANCE MODE.
 BIT #BIT8,#SRPTR ;CHECK STALL SWITCH
 BNE LTB ;BRANCH IF NO STALL WANTED
 STALL ;STALL
 LTB: TSTB #TPS ;WAIT FOR READY.
 BPL ;=4
 JSR X7,GTBINP ;GET BIN CHARACTER.
 MOVB X1,CRBUF+1 ;MOVE TO S/B CHAR.
 MOVB X1,PTPB ;OUTPUT BIN CHARACTER.
 TSTB #TKS ;WAIT FOR READER DONE.
 BPL ;=4
 MOVB #TKB,CRBUF ;CHAR IN READ BUFFER TO CRBUF.
 DATCHK ;GO CHECK AGAINST S/B CHAR.
 TST #0#42 ;CHAIN OR AUTO ACCEPT?
 BEQ LTA ;BR IF NOT.
 DEC CTRA ;DONE REQUIRED TIMES?
 LTA ;BR IF NOT.
 JMP CHNC ;YES, GO EXIT.

3251	014162	047	137	127	A33WP61	.BYTE	047,137,127,057,127,137
3252	014165	057	127	137			
3253	014170	047	133	077	A35WP61	.BYTE	047,133,077,103,077,133
3254	014173	103	077	133			
3255	014176	101	102	103	A1	.BYTE	101,102,103
3256	014201	104	105	106	D1	.BYTE	104,105,106
3257	014204	107	110	111	G1	.BYTE	107,110,111
3258	014207	112	113	114	J1	.BYTE	112,113,114
3259	014212	115	116	117	M1	.BYTE	115,116,117
3260	014215	120	121	122	P1	.BYTE	120,121,122
3261	014220	123	124	125	S1	.BYTE	123,124,125
3262	014223	126	127	130	V1	.BYTE	126,127,130
3263	014226	131	132	060	Y1	.BYTE	131,132,060
3264	014231	061	062	063	ONE1	.BYTE	061,062,063
3265	014234	064	065	066	FOUR1	.BYTE	064,065,066
3266	014237	067	070	071	SEVEN1	.BYTE	067,070,071
3267	014242	041	042	043	C411	.BYTE	041,042,043
3268	014245	044	045	046	C441	.BYTE	044,045,046
3269	014250	047	050	051	C471	.BYTE	047,050,051
3270	014253	052	053	054	C521	.BYTE	052,053,054
3271	014256	055	056	057	C551	.BYTE	055,056,057
3272	014261	072	073	074	C721	.BYTE	072,073,074
3273	014264	075	076	077	C751	.BYTE	075,076,077
3274	014267	100	133	134	C1001	.BYTE	100,133,134
3275	014272	135	136	137	C1351	.BYTE	135,136,137
3276	014275	377	000	377	C3771	.BYTE	377,000,377
3277	014300	021445	040524	020102	T0TST1	.ASCII	'*#TAB TESTX#'
3278	014306	042524	052123	021445			
3279	014314	020040	020040	020040	T0MRK1	.ASCII	' /0'
3280	014322	020040	040057				
3281	014326	020040	020040	020040	T0MRK11	.ASCII	' /0'
3282	014334	027440	100				
3283	014337	045	100		CRLF1	.ASCII	'*#'
3284	014341	055	026455	044455	RM33A1	.ASCII	'----I0'
3285	014346	100					
3286	014347	055	026511	100	RM33B1	.ASCII	'-I-0'
3287	014353	055	026455	044455	RM37A1	.ASCII	'----I-10'
3288	014360	044455	100				
3289	014363	134	040040		SPTSTC1	.ASCII	'\ 0'
3290	014366	021445	040503	051122	CRTST1	.ASCII	'*#CARRIAGE RETURN TESTX#'
3291	014374	040511	042507	051040			
3292	014402	052105	051125	020116			
3293	014410	042524	052123	021445			
3294	014416	100					
3295	014417	045	051043	043511	RMTST1	.ASCII	'*#RIGHT MARGIN TESTX#'
3296	014424	052110	046440	051101			
3297	014432	044507	020116	042524			
3298	014440	052123	021445	100			
3299	014445	045	051443	040520	SPTST1	.ASCII	'*#SPACE TESTX#'
3300	014452	042503	052040	051505			
3301	014460	022524	040043				
3302	014464	021445	044514	042516	LFTST1	.ASCII	'*#LINE FEED TESTX#'
3303	014472	043040	042505	020104			
3304	014500	042524	052123	021445			

3305	014506	100					
3306	014507	045	041443	040510	CHRTST1	.ASCII	'*#CHARACTER TESTX#'
3307	014514	040522	052103	051105			
3308	014522	052040	051505	051524			
3309	014530	021445	100				
3310	014533	045	053443	051117	WCPTST1	.ASCII	'*#WORST CASE PATTERN TESTX#'
3311	014540	052123	041440	051501			
3312	014546	020105	040520	052124			
3313	014554	051105	020116	042524			
3314	014562	052123	021445	100			
3315	014567	045	045443	041131	KMSG11	.ASCII	'*#KYBD TESTX#'
3316	014574	020104	042524	052123			
3317	014602	021445	100				
3318	014605	045	051120	051505	KMSG21	.ASCII	'*XPRESS A KEY WITHIN 10 SECONDS,0'
3319	014612	020123	020101	042513			
3320	014620	020131	044527	044124			
3321	014626	047111	030440	020060			
3322	014634	042523	047503	042116			
3323	014642	027123	100				
3324	014645	045	042443	044103	KMSG31	.ASCII	'*#ECHO TEST'
3325	014652	020117	042524	052123			
3326	014660	041445	040510	040322	KMSG3A1	.ASCII	'*#CHARACTER KEYED WILL BE TYPED,0'
3327	014666	052103	051105	045440			
3328	014674	054505	042105	053440			
3329	014702	046111	020114	042502			
3330	014710	052040	050131	042105			
3331	014716	056					
3332	014717	045	052522	047502		.ASCII	'*#RUBOUT ENDS ROUTINE,X#0'
3333	014724	052125	042440	042116			
3334	014732	020123	047522	052125			
3335	014740	047111	027105	021445			
3336	014746	100					
3337	014747	045	047443	052103	KMSG41	.ASCII	'*#OCTAL EQUIVALENT TEST0'
3338	014754	046101	042440	052521			
3339	014762	053111	046101	047105			
3340	014770	020124	042524	052123			
3341	014776	100					
3342	014777	045	040		KMSG51	.ASCII	'*#X 0'
3343	015001	040	020040	022440	OCTEQV1	.ASCII	'*#X 0'
3344	015006	100					
3345	015007	045	047516	045440	KMSG61	.ASCII	'*#NO KEYBOARD REQUEST,0'
3346	015014	054505	047502	051101			
3347	015022	020104	042522	052521			
3348	015030	051505	027124	100			
3349	015035	045	040506	051514	KMSG71	.ASCII	'*#FALSE KYBD INTERRUPT0'
3350	015042	020105	054513	042102			
3351	015050	044440	052116	051105			
3352	015056	052522	052120	100			
3353	015063	045	050043	044322	P7MG11	.ASCII	'*#PRINTER EXERCISERX#0'
3354	015070	052116	051105	042440			
3355	015076	042530	041522	051511			
3356	015104	051105	021445	100			
3357	015111	045	052043	050131	P7MG21	.ASCII	'*#TYPE IN DATA 10'
3358	015116	020105	047111	042040			

3359 015124 052101 020101 040072
3360 015132 020129 100
3361 015135 040 020040 020040 BKSU: ,ASCII 'U 0'
3362 015142 000001 DECVL: ,ASCII
DEND: ,END

A	014176	ACNV	004264	ACNVB	004220	ACNVC	004246
ACNVM	004300	ACNVX	004262	ACNV4	004226	ACNV6	004200
ADTENP	004526	ARDB	002412	AREAD	002364	AT20E	006052
A1ST	004254	A33WP6	014162	A35WP6	014170	BCHECK	003576
BDCNV	004420	BDCNVA	004440	BELL	000007	B1T0	000001
B1T1	000002	B1T10	002000	B1T11	004000	B1T12	010000
B1T13	020000	B1T14	040000	B1T15	100000	B1T2	000004
B1T3	000010	B1T4	000020	B1T5	000040	B1T6	000100
B1T7	000200	B1T8	000400	B1T9	001000	BKSU	015132
BLKBB	015265	BLKCC	015377	BLK2	015267	BLOCKA	015142
BLOCKB	015254	BLOCKC	015366	BLOCK1	015144	BLOCK2	015256
BMOVA	004350	BMOVE	004340	BRCRTR	001414	BREAD	002572
BREADA	002620	BREADB	002632	BREADC	002652	BREADD	002662
BSYNC	003636	CC	177776	CHAINN	001754	CHALT	104005
CHKASR	104022	CHK33	002210	CHK33A	002220	CHK33B	002224
CHK35	002226	CHLT	001422	CHNA	002004	CHNAA	002026
CHNB	002032	CHNC	002052	CHRTST	014507	CHR1	001366
CHR1A	001374	CHR2	001370	CHR2A	001376	CHR3	001372
CHR3A	001400	CKASR	002240	CK33	104013	CK35	104014
CK37	104015	CLEAN	001654	CNVCTR	004520	CPRDY	004362
CPRDYA	004372	CRBUF	001364	CRLF	014337	CRTA	001612
CRTB	001634	CRTST	014366	CTRA	001404	CTRB	001406
CTRC	001410	CTRD	001412	CT0A	010124	CT0B	010134
CT0C	010142	CT1A	010252	CT2A	010310	CT2B	010330
CT2C	010336	CT2D	010360	CT2E	010416	CT3A	010464
CT3B	010514	CT33A	011344	CT34A	011426	CT4A	010550
CT4B	010562	CURTST	001236	C100	014267	C1112	013726
C1112A	013746	C1112M	013720	C135	014272	C377	014275
C41	014242	C44	014245	C47	014250	C52	014253
C55	014256	C72	014261	C75	014264	D	014201
DATCHK	104004	DATHLT	104017	DECVL	015135	DELAY	104024
DELAYX	104400	DEND	015142	DIGIT	004522	DLCNT	003050
DLY	003046	DLYA	003070	DLYB	003076	DLYR	003304
DLYR0	003350	DLYR1	003362	DLYT	003144	DLYX	003344
DLYXA	003356	DLYXB	003364	DTCHK	001446	DTCHKA	001464
DYFL	013330	DYFLA	013342	DYFLB	013354	DTHLT	001456
DVDND	001416	DVQUOT	001420	EHALT	104010	EHLT	001434
EMTINT	002170	EMTTAB	001304	EMTX	000027	ERCTR	001402
ERR	001466	ERRA	001506	ERROR	104003	ET0A	012146
ET0B	012204	ET0C	012220	ET0CA	012226	ET1A	012250
ET2A	012340	FBALL	004746	F0F3	004702	F0F3A	004712
F0F3B	004720	FORWD	002112	FORWDA	002144	FORWDB	002152
FOUR	014234	FW336	005006	FW356	005046	G	014204
GETRDY	001646	GRCNT	005106	GRCNT	003552	GTA	013416
GTB	013434	GTBIN	004064	GTBINP	004132	GTRDYA	001676
GTRDYB	001702	GTRDYC	001716	GTRDYD	001740	HERE	002076
HTA	013464	HTB	013522	HTC	013600	HTD	013612
ICTR	001244	INBIN	004026	INCPRO	001606	INCRTN	001750
INIT0	013280	INIT1	013304	ITA	013664	J	014207
JTA	013674	KMSG1	014567	KMSG2	014605	KMSG3	014645
KMSG3A	014660	KMSG4	014747	KMSG5	014777	KMSG6	015007
KMSG7	015035	KSTART	001234	KTA	013764	KT0	013772
LFTST	014464	LOGIC	002062	LSPCH	004376	LTA	014062
LTB	014102	M	014212	MACHER	000004	MANUAL	100000

HSEC	003072	NOP	000240	NXTST	001242	OCTEQV	013001
ONE	014231	OPEN	000000	P	014215	PC	0000007
PCHCNT	012512	PCHDAT	012734	PCHDTA	012766	PCHECK	012014
PCHKA	012022	PCHKB	012030	PCHKC	012062	PCHKD	012100
PCHZER	013024	PCONT	012532	PCONTA	012564	PCONTB	012000
PCONTC	012012	PRFNT	011730	PIND	004000	PLTLR	011766
PLTRA	011774	POPAP	005726	POPBP2	022626	PRGID	001250
PRGNUM	001832	PRDTAB	001252	PRG0	005144	PRG1	007652
PRG10	013626	PRG10A	013634	PRG10B	013690	PRG11	013662
PRG12	013672	PRG13	013762	PRG14	014034	PRG2	010026
PRG3	011462	PRG4	012110	PRG5	012410	PRG6	013370
PRG7	013454	PRTY0	000000	PRTY1	000040	PRTY2	000100
PRTY3	000140	PRTY4	000200	PRTY5	000240	PRTY6	000300
PRTY7	000340	PSH	177776	PT0	004052	PTOP	004060
PT1	004054	PT1P	004062	P0A	005174	P0AAA	006776
P0AAB	007020	P0AAE	007026	P0AE	005202	P0BA	005224
P0BAA	007046	P0BAC	007100	P0BE	005232	P0CA	005254
P0CAA	007116	P0CAC	007152	P0CAD	007166	P0CAE	007172
P0CAF	007174	P0CE	005262	P0DA	005304	P0DAA	007216
P0DAB	007232	P0DE	005312	P0EA	005334	P0EAA	007276
P0EAB	007322	P0EAC	007332	P0EB	005356	P0EC	005376
P0FA	005416	P0FAA	007334	P0GA	005454	P0GAA	007416
P0GAB	007466	P0HA	005516	P0HAA	007506	P0HAB	007464
P0IA	005544	P0JA	005600	P0JAA	007600	P0KA	005634
P0KAA	007630	P0KB	005642	P0KC	005662	P0LA	005676
P0MA	005760	P0MB	006004	P0NA	006032	P0NC	006054
P0OA	006102	P0OE	006132	P0PA	006162	P0PB	006214
P0QA	006242	P0OC	006276	P0QD	006312	P0QE	006316
P0RA	006344	P0SA	006426	P0TA	006462	P0TLST	177777
P0T0	005156	P0T1	005206	P0T10	005530	P0T11	005564
P0T12	005622	P0T13	005664	P0T14	005732	P0T15	006006
P0T16	006056	P0T17	006136	P0T2	005236	P0T20	006222
P0T21	006326	P0T22	006410	P0T23	006452	P0T24	006526
P0T25	006562	P0T26	006604	P0T27	006634	P0T3	005266
P0T30	006670	P0T31	006722	P0T32	006762	P0T33	007032
P0T34	007106	P0T35	007202	P0T36	007260	P0T37	007342
P0T4	005316	P0T40	007402	P0T41	007472	P0T42	007570
P0T43	007620	P0T5	005400	P0T6	005442	P0T7	005474
P0UA	006536	P0VA	006572	P0WA	006614	P0XA	006644
P0YA	006700	P0ZA	006736	P0ZB	006760	P1AA	007722
P1BA	007750	P1CA	010000	P1CC	010006	P1TLST	177777
P1T0	007706	P1T1	007734	P1T2	007764	P2TLST	177777
P2T0	010060	P2T1	010206	P2T10	010774	P2T11	011006
P2T12	011020	P2T13	011032	P2T14	011044	P2T15	011056
P2T16	011070	P2T17	011102	P2T2	010272	P2T20	011114
P2T21	011126	P2T22	011140	P2T23	011152	P2T24	011164
P2T25	011176	P2T26	011210	P2T27	011222	P2T3	010426
P2T30	011234	P2T31	011246	P2T32	011260	P2T33	011316
P2T34	011400	P2T4	010520	P2T5	010732	P2T6	010750
P2T7	010762	P3AA	011520	P3AB	011536	P3BA	011976
P3BB	011614	P3CA	011664	P3CB	011702	P3CC	011710
P3CD	011736	P3TLST	177777	P3T0	011510	P3T1	011566
P3T2	011646	P4TLST	177777	P4T0	012134	P4T1	012240
P4T2	012320	P7MG1	015063	P7MG2	015111	RBUSY	012510

RCMSK	003572	RCNT	001362	RCONT	012644	RCONTA	012666
RCONTB	012714	RCONTC	012726	RDAT	013132	RDATA	013136
RDATB	013172	RDATC	013226	RDELAY	104025	RDLCNT	003210
RDLY	003206	RDLYA	003230	RDLYB	003236	RESET2	104023
RIND	004050	RMB	010266	RMTST	014417	RM33A	014341
RM33B	014347	RM37A	014353	RNCNT	003574	RNGEN	002520
RP1	002566	RP2	002570	RSETT2	002512	RSTAL	003306
RSTALL	104026	RSTLA	003320	RSTLAA	003340	RSTLB	003342
RSTPC	002360	RSTPSW	002362	RSTREG	104021	RSTRG	002324
RYNNO	001240	RZERA	013104	RZERO	013064	R0	0000000
R1	0000001	R2	0000002	R2OZER	013230	R3	0000003
R4	0000004	R5	0000005	R6	0000006	S	014220
SAVREG	104020	SAVRG	002264	SCOPE	104012	SCOPT	001246
SEED0	013272	SEED1	013316	SEVEN	014237	SHALT	001510
SHLTA	001526	SOFTSR	000176	SP	0000006	SPBOT	001200
SPCNT	010630	SPTST	014445	SPTSTC	014363	SRESET	104011
SRPTR	000174	SRSET	001644	SRSETT	002474	STAL	003146
STALA	003200	STALAA	003160	STALB	003202	STALL	104002
START	001530	STBF	004634	STBFA	004650	STLMSK	003204
STLSPV	002444	STLSRV	002414	STPCHV	104007	STPPA	002462
STPRA	002432	STRDRV	104006	SUBTEN	004460	SUBTNA	004464
SUBTNB	004500	SVRPC	002320	SVRPSW	002322	SVCTRA	004024
SYNCA	003716	SYNCB	003724	SYNCC	004012	TABP	010632
TABPA	010644	TABPB	010654	TABPC	010672	TBCNT	010626
TBMRK	014314	TBMRK1	014326	TBTST	014300	TCTR	004572
TENPWR	004524	TIMCAL	003402	TIMCLA	003430	TIMCLB	003446
TKB	001212	TKLVL	001222	TKS	001210	TKVTR	001220
TPB	001216	TPBM	010602	TPBMA	010612	TPLVL	001226
TPL3A	004616	TPS	001214	TPVTR	001224	TSPCH	003456
TSPCHA	003542	TSPCHB	003550	TSPCHC	003500	TSTPCH	012514
TSTRDR	012620	TTYTYP	001230	TYP	002666	TYPA	002676
TYPC	002720	TYPD	002746	TYPDAT	003012	TYPE	104000
TYPE5	104001	TYPF	002764	TYPG	002776	TYPLA	004546
TYPLB	004552	TYPLN	004540	TYPLN3	104016	TYPL3	004574
TYPS	003014	TYPSA	003040	TYPSB	003042	V	014223
WCPTST	014533	X	000002	XTY	013756	Y	014226
Z	000004	.	015142				

ERRORS DETECTED: 0

*DZKLAE,DZKLAE=DZKLAE./SOL
 RUN-TIME: 16 27 0 SECONDS
 CORE USED: 15K