

This microfiche card contains a grid of 14 columns and 14 rows of tiny document images. The images are too small to read but appear to be technical diagrams or test results. The grid is located on the left side of the card, with the right side being a large, dark, blank area.

1. ABSTRACT

THE TYPESET-11 READER AND PUNCH TESTS CONSISTS OF A PACKAGE OF TEST PROGRAMS DESIGNED TO TEST THE PA611 READER LOGIC, READER, PUNCH LOGIC, PUNCH, AND THE READER AND PUNCH IN COMBINATION. ALL TESTS ARE INCLUDED IN ONE OBJECT TAPE.

THE AVAILABLE TESTS ARE LISTED HERE IN NUMERICAL ORDER:

PRG0 -READER TEST
 PRG1 -PUNCH TEST
 PRG2 -PUNCH VERIFY ROUTINE
 PRG3 -COMBINED READER-PUNCH TEST
 PRG4 -PUNCH TAPE WITH 2 CHARACTERS SET IN SR ROUTINE.
 PRG5 -READ AND CHECK TAPE PUNCHED WITH 2 CHARACTERS SET IN SR.
 PRG6 -READ X CHARACTERS, THEN STALL Y MSECS.
 PRG7 -SPECIAL BINARY COUNT PATTERN TAPE GENERATOR.
 PRG10-READER SPEED PRINT ROUTINE.
 PRG11-PUNCH SPEED PRINT ROUTINE.
 PRG12-PUNCH LOGIC INIT TEST
 PRG13-READER LOGIC LIGHT TEST

PROGRAMS PRG0 THROUGH PRG3 ARE THE READER AND PUNCH TESTS. PROGRAMS PRG4 THROUGH PRG11 ARE UTILITY ROUTINES THAT PRODUCE TEST TAPES AND AID IN MAKING ADJUSTMENTS.

2. REQUIREMENTS2.1 EQUIPMENT

- A. PDP-11 PROCESSOR. (4K CORE)
- B. ASR33/35 TELETYPE
- C. PA611 READER(S) AND PUNCH(S).

THE PROCESSOR AND TELETYPE MUST BE IN OPERATING CONDITION.

THE TELETYPE MUST BE AT ITS STANDARD PERIPHERAL ADDRESSES. SEE SECTION 7.3 FOR OPERATION WITH NON-STANDARD PERIPHERAL ADDRESSES.

2.2 STORAGE

THIS PROGRAM USES LOCATION 00200 THROUGH 015600.

3. LOADING PROCEDURE

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

3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107

THE PROGRAM WILL SELF-START IN ORDER TO INITIALIZE FOR THE NUMBER OF READERS AND PUNCHES IN THE SYSTEM. FOLLOW TYPED INSTRUCTIONS.

3.1 RESTART ADDRESS

THE RESTART ADDRESS OF THIS PROGRAM IS LOCATION 1004.

RESTART ADDRESS: 1004

3.2 START PROCEDURE

IN GENERAL, ALL TESTS ARE INITIATED BY LOADING ADDRESS 200, DEPOSITING TEST # IN SWITCH REGISTER, AND HITTING CONTINUE.

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
1634. USE PROCEDURE4.1 PRG0 USE PROCEDURE (DESCRIPTION IN SECTION 8.1)

- A. INSURE THAT TELETYPE IS ON-LINE
- B. LOAD READER WITH SPECIAL BINARY COUNT PATTERN TEST LOOP. IF NOT USING A LOOP, DATA MUST BE UNDER READ HEAD.
- C. LOAD ADDRESS 00200.
- D. SET SR TO 00000. PRESS START.
- E. THE PROGRAM IDENTIFIES ITSELF, TYPES SET UP AND SR OPTION INSTRUCTIONS. SELECT ANY DESIRED OPTIONS.

SR15=1 HALT ON ERROR.
 SR14=1 ENTER SCOPE MODE.
 SR13=1 INHIBIT ERROR PRINT.
 SR11=1 INHIBIT ITERATION.
 SR10=1 HALT AT END OF CURRENT ROUTINE.
 SR9=1 SELECT A SPECIFIC ROUTINE FOR EXECUTION.
 SR7 THROUGH SR0=NUMBER OF ROUTINE TO BE SELECTED.

SECTION 7.2 GIVES A COMPLETE EXPLANATION OF SR OPTIONS.

- F. PRESS CONTINUE. FOLLOW PROGRAM INSTRUCTIONS
- G. REFER TO SECTION 6.2 IF ANY ERROR PRINTOUTS OCCUR.
- H. WHEN THE PROGRAM HAS COMPLETED ONE PASS IT WILL TYPE P00END. ONE ERROR-FREE PASS WILL TAKE APPROXIMATELY 10 MINUTES.

4.2 PRG1 USE PROCEDURE (DESCRIPTION IN SECTION 8.2)

- A. INSURE THAT TELETYPE IS ON-LINE.
- B. INSURE THAT THE PUNCH HAS AN ADEQUATE SUPPLY OF TAPE.
- C. LOAD ADDRESS 00200.
- D. SET SR TO 000001. PRESS START.
- E. THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO SET ANY DESIRED SR OPTIONS. SELECT ANY DESIRED OPTIONS.

SR15=1 HALT ON ERROR.
 SR14=1 ENTER SCOPE MODE.
 SR13=1 INHIBIT ERROR PRINT.
 SR11=1 INHIBIT ITERATION.
 SR10=1 HALT AT END OF CURRENT ROUTINE.
 SR9=1 SELECT A SPECIFIC ROUTINE FOR EXECUTION.
 SR7 THROUGH SR0=NUMBER OF ROUTINE TO BE SELECTED.

A FULL EXPLANATION OF SR OPTIONS IS GIVEN IN SECTION 7.2.

- F. PRESS CONTINUE. FOLLOW PROGRAM INSTRUCTIONS.
- G. UPON COMPLETION OF A PROGRAM PASS THE PROGRAM WILL TYPE "P01 END".
- H. REFER TO SECTION 6. IF ANY ERRORS OCCUR.

F01

PA611 MACY11 27(1006) 04-NOV-76 12:14 PAGE 5
DZPAAA.CMB 04-NOV-76 12:11

164
165

ONE ERROR-FREE PASS WILL TAKE APPROXIMATELY 10 MINUTES.

166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
2214.3 PRG2 USE PROCEDURE (DESCRIPTION IN SECTION 8.3)

- A. INSURE THAT TELETYPE IS ON-LINE.
- B. LOAD TAPE THAT WAS PUNCHED BY PRG1-PUNCH TEST IN READER.
LOAD TAPE SO THAT THE FIRST RUBOUT CHARACTER (ALL 1'S) IS ON THE RIGHT EDGE OF THE METAL PLATE OVER THE READ STATION. MAKE READER READY.
- C. LOAD ADDRESS 00200.
- D. SET SR TO 000002. PRESS START.
- E. THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO LOAD THE READER.
- F. PRESS CONTINUE. THE PROGRAM WILL READ THE TAPE AND REPORT ANY ERRORS. DISREGARD ANY ERRORS THAT OCCUR WHEN THE READER REACHES THE END OF THE TAPE.
- G. THE SR OPTIONS AVAILABLE IN THIS PROGRAM ARE:

SR15=1 HALT ON ERROR.
SR13=1 INHIBIT ERROR PRINT.

- H. REFER TO SECTION 6. IF ERRORS OCCUR.

PRG2 DOES NOT RESYNC THE READER AT ANY TIME. IT'S INTENT IS TO SHOW EACH AND EVERY ERROR CAUSED BY THE PUNCH.

EXECUTION TIME DEPENDS ON LENGTH OF TAPE TO BE VERIFIED.

4.4 PRG3 USE PROCEDURE (DESCRIPTION IN SECTION 8.4)

- A. INSURE THAT TELETYPE IS ON-LINE.
- B. INSURE THAT THE PUNCH HAS AN ADEQUATE SUPPLY OF TAPE.
- C. USING THE "PUNCH FEED" KEY, PUNCH 2 FEET BLANK LEADER.
LOAD A 1" THICK STACK OF PREPUNCHED SPECIAL BINARY COUNT PATTERN TAPE IN READER, AND MAKE READER READY. THE BLANK LEADER PORTION OF THE TAPE MUST BE AT THE READ STATION.
- D. LOAD ADDRESS 000200.
- E. SET SR TO 000003. PRESS START.
- F. THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO PUNCH LEADER AND LOAD READER.
- G. PRESS CONTINUE. THE PROGRAM WILL PUNCH A NEW BINARY COUNT PATTERN WHILE READING THE PREPUNCHED TAPE IN THE READER. THE PROGRAM SHOULD RUN ERROR-FREE UNTIL THE READER TAPE IS EXHAUSTED, AT WHICH POINT A READER NOT READY MESSAGE WILL OCCUR. REPLACE THE READER TAPE WITH THE TAPE JUST PUNCHED AND RERUN THE TEST. RUN THE TEST 6 TIMES.
- H. THE SR OPTIONS AVAILABLE WITH THIS PROGRAM ARE:

SR15=1 HALT ON ERROR.
SR13=1 INHIBIT ERROR PRINT.

- I. REFER TO SECTION 6. IF ERRORS OCCUR.

EXECUTION TIME: PRG3 IS CONTINUOUS RUNNING.

HO1

PA511 MACY11 27(1006) 04-NOV-76 12:14 PAGE 7
DZPAAA.CMB 04-NOV-76 12:11

222
223

224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266

4.5 PRG4 USE PROCEDURE (DESCRIPTION IN SECTION 8.5)

THIS PROGRAM CONTINUOUSLY PUNCHES TAPE WITH 2 CHARACTERS
 WHOSE CODES HAVE BEEN SET IN SR. TO RUN:

- A. INSURE THAT TELETYPE IS ON-LINE.
- B. INSURE TAHT THE PUNCH HAS AN ADEQUATE SUPPLY OF TAPE.
- C. LOAD ADDRESS 000200
- D. SET SR TO 000004. PRESS START.
- E. THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO SET SR TO DESIRED CODES AND PUNCH READY.
- F. PRESS CONTINUE. THE PROGRAM WILL PUNCH THE DESIRED CHARACTERS CONTINUOUSLY UNTIL STOPPED BY USER.
- G. THE CHARACTERS TO BE PUNCHED MAY BE CHANGED WHILE THE PROGRAM IS RUNNING.
- H. THIS PROGRAM HAS NO SR OPTIONS.

EXECUTION TIME: CONTINUOUS RUNNING PROGRAM.

4.6 PRG5 USE PROCEDURE (DESCRIPTION IN SECTION 8.6)

THIS PROGRAM READS AND CHECKS A TAPE PUNCHED WITH ANY 2 CHARACTERS
 WHOSE CODES HAVE BEEN SET IN THE SR. TO RUN:

- A. INSURE THAT TELETYPE IS ON-LINE.
- B. LOAD TAPE TO BE READ IN READER. DATA MUST BE UNDER READ STATION.
- C. LOAD ADDRESS 000200.
- D. SET SR TO 000005. PRESS START.
- E. FOLLOW PROGRAM INSTRUCTIONS.
- F. THE PROGRAM WILL READ THE TAPE AND REPORT ANY ERRORS.
- G. THE SR OPTIONS AVAILABLE WITH THIS PROGRAM ARE:

SR15=1 HALT ON ERROR.
 SR13=1 INHIBIT ERROR PRINT.

- H. REFER TO SECTION 6. IF ERRORS OCCUR.

EXECUTION TIME: CONTINUOUS RUNNING PROGRAM.

267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311

4.7 PRG6 USE PROCEDURE

THIS PROGRAM IS INTENDED AS AN AID IN SCOPING AND ADJUSTING THE READER AND READER LOGIC. TO RUN:

- A. INSURE THAT TELETYPE IS ON-LINE.
- B. LOAD ANY TAPE LOOP IN THE READER. ONE'S AND ZEROES LOOP IS A GOOD CHOICE.
- C. LOAD ADDRESS 000200
- D. SET SR TO 000006. PRESS START
- E. THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO SET THE SR TO NUMBER OF CHARACTERS TO READ AND TO NUMBER OF MILLISECONDS TO STALL AFTER READING THE CHARACTERS. PLEASE NOTE:
 - 1. THE LEFT 8 SWITCHES ARE FOR THE NUMBER OF CHARACTERS TO BE READ. THE RANGE IS BETWEEN 1 AND 377(8).
 - 2. THE RIGHT 8 SWITCHES ARE FOR SETTING THE NUMBER OF MILLISECONDS TO STALL AFTER READING THE NUMBER OF CHARACTERS SPECIFIED.
- F. PRESS CONTINUE. THE PROGRAM WILL CONTINUOUSLY READ AND STALL UNTIL STOPPED BY USER.
- G. THE SETTINGS OF THE SR MAY BE CHANGED AT ANY TIME.

EXECUTION TIME: CONTINUOUS RUNNING PROGRAM.

4.8 PRG7 USE PROCEDURE

THIS PROGRAM CONTINUOUSLY PUNCHES A TAPE WITH THE SPECIAL BINARY COUNT PATTERN. TO RUN:

- A. INSURE THAT TELETYPE IS ON-LINE
- B. MAKE SURE THAT THE PUNCH HAS AN ADEQUATE SUPPLY OF TAPE.
- C. LOAD ADDRESS 000200.
- D. SET SR TO 000007. PRESS START
- E. THE PROGRAM IDENTIFIES ITSELF, AND TYPES INSTRUCTION TO MAKE THE PUNCH READY.
- F. PRESS CONTINUE. THE SPECIAL BINARY COUNT PATTERN WILL BE PUNCHED UNTIL THE PROGRAM IS STOPPED BY USER.

4.9 PRG10 USE PROCEDURE

THIS PROGRAM IS INTENDED AS AN AID IN DETERMINING THE SPEED OF THE READER. IT IS NOT INTENDED TO REPLACE REGULAR SCOPING PROCEDURES FOR SETTING THE READER TO ITS CORRECT SPEED.

WITH THIS PROGRAM THE READER SPEED CAN BE MEASURED IN TWO WAYS:

1. 30 SECOND MEASUREMENT PERIOD. PLUS OR MINUS 10 CHARACTER ACCURACY
2. 300 SECOND (5 MINUTE) MEASUREMENT PERIOD. PLUS OR MINUS 1 CHARACTER ACCURACY

IN EITHER CASE MEASUREMENT ACCURACY DEPENDS ON THE USER'S ATTENTION TO STARTING AND ENDING TIMES OF MEASUREMENT, AS THE TIME INTERVALS ARE DETERMINED BY THE USER USING A SWEEP SECOND HAND WATCH OR STOP WATCH.

THE SPECIFIED ACCURACY ASSUMES THAT THE USER WILL TERMINATE THE MEASURING INTERVAL WITHIN ONE SECOND OF THE MEASUREMENT PERIOD. TO RUN:

- A. INSURE THAT TELETYPE IS ON-LINE
- B. MOUNT ANY TAPE LOOP IN READER.
- C. LOAD ADDRESS 000200.
- D. SET SR TO 000010. PRESS START
- E. THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO LOAD READER AND MAKE READY, AND TO SELECT DESIRED MEASUREMENT PERIOD.
- F. PRESS CONTINUE WHEN READY TO START MEASUREMENT. THE READER WILL START RUNNING.
- G. AT END OF TIME PERIOD, SET SR15 TO A 1 AND BACK TO 0 AGAIN. THE PROGRAM WILL TYPE READER SPEED IN CHARACTERS PER SECOND AND HALT.
- H. TO REPEAT, SELECT THE DESIRED TIME PERIOD WITH SR14, MAKE SURE THAT SR15 IS SET TO 0, AND PRESS CONTINUE WHEN READY.

4.10 PRG11 USE PROCEDURE

THIS PROGRAM IS INTENDED AS AN AID IN DETERMINING THE PUNCH SPEED. THE SPEED OF THE PUNCH CAN BE MEASURED WITHIN ONE CHARACTER ACCURACY PROVIDED THE USER PAYS CLOSE ATTENTION TO THE STARTING AND STOPPING TIME OF THE MEASUREMENT PERIOD. THE MEASUREMENT PERIOD IS CONTROLLED BY THE USER USING A SWEEP SECOND WATCH OR STOP WATCH. THE PERIOD USED IS 60 SECONDS. TO RUN:

- A. INSURE THAT TELETYPE IS ON-LINE
- B. INSURE THAT PUNCH HAS AN ADEQUATE SUPPLY OF TAPE.
- C. LOAD ADDRESS 000200.
- D. SET SR TO 000011. PRESS START
- E. THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO MAKE PUNCH READY.
- F. PRESS CONTINUE WHEN READY TO START. THE PUNCH WILL START RUNNING.
- G. AT END OF TIME PERIOD (60 SECONDS), SET SR15 TO A 1 AND BACK TO 0. THE PROGRAM WILL TYPE PUNCH SPEED IN CHARACTER PER SECOND AND HALT.
- H. TO REPEAT, MAKE SURE THAT SR15 IS SET TO A 0, AND PRESS CONTINUE.

312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367

368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
4234.11 PRG12 USE PROCEDURE

THIS PROGRAM IS INTENDED TO BE USED AS AN AID IN CHECKING OUT THE ADDITIONAL LOGIC ADDED TO THE PUNCH CONTROLLER THAT ALLOWS THE PROGRAMMER TO ISSUE "PUNCH RESET" UNDER SOFTWARE CONTROL.

THIS CODE EXECUTES UNDER OPERATOR INTERVENTION. THE OPERATOR TRIES TO "HANG" THE PUNCH BY PERFORMING THE FOLLOWING STEPS.

0. FIRST MAKE SURE THE PUNCH IS READY. THE PUNCH KNOB SHOULD BE SET TO THE "AVAILABLE" POSITION. RUN THIS TEST THE PUNCH WILL START PUNCHING A BINARY BY LOADING ADDRESS BY LOADING ADDRESS 200, SETTING THE SWITCH REGISTER TO 12 AND HITTING THE START SWITCH. THE PUNCH WILL START PUNCHING A BINARY PATTERN.
1. REMOVE THE PAPER ROLL FROM IT'S HOLDER AND PLACE IT ON THE RIGHT HAND SIDE OF THE PUNCH. (DO NOT TEAR THE PAPER TAPE AND MAKE SURE IT FEEDS CORRECTLY). THE TEST WILL CONTINUE TO PUNCH OUT AN ENDLESS STREAM OF CHARACTERS.

THIS STREAM OF CHARACTERS CAN ONLY BE INTERRUPTED BY "HANGING" THE PUNCH CONTROL LOGIC.

2. TO HANG THE CONTROL LOGIC, TURN THE PUNCH "OFF" VIA THE "OFFLINE/ONLINE" KNOB ON THE TOP OF THE PUNCH, WHILE CHARACTERS ARE BEING PUNCHED OUT. WAIT FOR THE PUNCH MOTOR TO STOP.

THIS WILL CAUSE THE PUNCH TO SHUTOFF.

5. AT THIS POINT

THE PUNCH SHOULD BE HUNG. TO SEE IF THE PUNCH IS HUNG, TURN THE CONTROL KNOB BACK TO "AVAILABLE". IF TURNING THE KNOB BACK TO AVAILABLE CAUSES MORE CHARACTERS TO BE PUNCHED, THEN THE PUNCH IS NOT HUNG. IF CHARACTERS ARE BEING PUNCHED OUT, THEN THIS SHUT- OFF PROCESS MUST BE REPEATED UNTIL THE PUNCH HANGS. IF THE PUNCH DOES NOT HANG, START AGAIN AT STEP #2.

AFTER THE PUNCH IS HUNG, TURN THE CONTROL KNOB BACK TO "AVAILABLE".

6. IF THE PUNCH DOES HANG, THEN THE PROGRAMMABLE INIT FUNCTION IS READY TO BE TESTED. HITTING "CONTINUE" ON THE PDP-11 CONTROL PANEL WILL CAUSE THE PROGRAMMABLE INIT TO BE INVOKED. THE PROGRAM WILL CONTINUE TO PUNCH CHARACTERS IF THE INIT IS WORKING CORRECTLY. IF THE PUNCH IS STILL HANGING THEN THE PROGRAMMABLE INIT DID NOT WORK.

MO1

PA611 MACY11 27(1006) 04-NOV-76 12:14 PAGE 12
DZPAAA.CMB 04-NOV-76 12:11

424
425
426
427
428

7. SWITCH REGISTER OPTIONS:

SR14=1 TO SCOPE LOOP ON THE PROGRAMMABLE INIT FUNCTION

429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
4754.12 PRG13 USE PROCEDURE

THIS PROCEDURE IS USED TO DETERMINE IF THE READER LIGHT LOGIC IS WORKING CORRECTLY. IT CHECKS THE ABILITY OF THE READER LIGHT TO BE TURNED OFF UNDER PROGRAM CONTROL.

STEPS:

RUN THIS TEST BY SELECTING IT VIA THE SWITCH REGISTER AND STARTING AT LOCATION 200.

1. PUT SWITCH 8 OF THE SWITCH REGISTER TO ZERO. PUT THE READER NUMBER IN THE SWITCH REGISTER WHEN THE PROGRAM ASKS FOR IT.
2. AFTER SELECTING A GIVEN READER VIA THE SWITCH REGISTER, TURN THE READER LIGHT "ON" BY PRESSING THE MOMENTARY CONTACT "ON/OFF" SWITCH ON THE READER.

IF THE READER LIGHT DOES NOT COME ON, THEN EITHER THE LIGHT OR THE SWITCH IS PROBABLY DEFECTIVE.
3. ONCE THE READER LIGHT IS ON, THE OPERATOR SHOULD PUT SWITCH 8 TO A ONE TO TURN IT OFF. THE PROGRAM MONITORS THE POSITION OF SWITCH 8. WHEN SWITCH 8 IS ONE, THE PROGRAMS ISSUES A SOFTWARE COMMAND TO TURN THE LIGHT OFF. IF THE LIGHT REMAINS ON AFTER SWITCH 8 HAS BEEN SET TO A ONE, THEN THE READER LOGIC IS DEFECTIVE.
4. IF THE READER LIGHT GOES OFF WHEN SWITCH 8 IS IN THE ONE STATE, THEN THE TEST HAS WORKED CORRECTLY.

TO SELECT A NEW READER, PUT SWITCH 12 TO A ONE AND HIT CONTINUE.

TO RE-RUN THE TEST, GO DO STEP 1 AND STEP 2, THEN HIT CONTINUE.
5. SWITCH REGISTER OPTIONS AVAILABLE IN THIS TEST ARE:

SR12=1	TO SELECT A NEW READER TO TEST.
SR 8=1	TO TURN LIGHT OFF ON READER SELECTED.
SR14=1	TO DO A SCOPE LOOP ON THE PROGRAMMABLE READER LIGHT DISABLE FUNCTION.

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

5. PROGRAM AND/OR OPERATOR ACTION

5.1 NORMAL HALTS

LOC 002502 COMMON HALT. THIS HALT IS CONTAINED IN A SUBROUTINE THAT IS CALLED BY THOSE PARTS OF THE PROGRAM REQUIRING THAT THE PROCESSOR HALT. THIS HALT NORMALLY OCCURS UPON COMPLETION ON AN INSTRUCTION OR STATUS PRINTOUT. WHEN THE HALT OCCURS, THE CONSOLE DATA LIGHTS DISPLAY THE ADDRESS OF INSTRUCTION THAT GENERATED THE HALT REQUEST.

LOC 002032 ROUTINE END HALT. THIS HALT OCCURS UPON COMPLETION OF THE CURRENT TEST ROUTINE. WHEN THE HALT OCCURS, THE CONSOLE DATA LIGHTS DISPLAY THE NUMBER OF ROUTINE JUST COMPLETED. THE HALT OCCURS ONLY IF SR10 IS SET TO A 1, FOR THOSE PROGRAMS THAT MAKE USE OF THE OPTION (PRGO, PRG1).

5.2 NORMAL PRINTOUTS

NORMAL PRINTOUTS IN THIS PROGRAM SERVE TO IDENTIFY A STARTING PROGRAM, TO PROVIDE INSTRUCTIONS, TO INDICATE STATUS, OR TO SIGNAL AN OPERATOR ERROR. MOST PRINTOUTS ARE SELF-EXPLANATORY. THOSE PRINTOUTS REQUIRING ADDITIONAL EXPLANATION FOLLOW.

"?INVALID PROGRAM"

THE USER HAS SELECTED FOR EXECUTION A NON-EXISTENT PROGRAM. SET IN SR3 THROUGH SR0 THE CORRECT PROGRAM NUMBER AND PRESS CONTINUE.

"?INVALID TEST"

THE USER HAS SELECTED FOR EXECUTION A NON-EXISTENT ROUTINE. SET CORRECT ROUTINE NUMBER IN SR7 THROUGH SR0 AND PRESS CONTINUE.

"PXX END."

THE SPECIFIED PROGRAM HAS COMPLETED ONE PASS.

6. ERRORS

ERRORS ARE REPORTED IN THIS PROGRAM BY ONE OF THE FOLLOWING METHODS:

- A. UNCONDITIONAL ERROR HALTS, OR
- B. ERROR PRINTOUT FOLLOWED BY AN OPTIONAL ERROR HALT.

6.1 UNCONDITIONAL ERROR HALTS

AN UNCONDITIONAL 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 000775.

- 000002 - RESERVED AREA.
- 000006 - ERROR TRAP
- 000012 - RESERVED INSTRUCTION TRAP
- 000016 - DEBUG TRAP
- 000022 - IOT TRAP
- 000026 - POWER FAIL TRAP
- 000040 THROUGH 000776 - SYSTEM SOFTWARE AND INTERRUPT VECTOR AREA, EXCEPT FOR READERS, PUNCHES, AND TTY VECTORS.

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

- A. EXAMINE 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 PROGRAM LISTING THE DISPLAYED PC VALUE.
- E. THE INSTRUCTION THAT IMMEDIATELY PRECEDES THE ONE REFERENCED BY THE DISPLAYED PC VALUE IS THE INSTRUCTION THAT WAS/WAS BEING EXECUTED WHEN THE FAILURE OCCURRED.

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

6.2 ERROR PRINTOUTS

ERROR PRINTOUTS IN THIS PROGRAM CAN BE ONE OF TWO TYPES:

- A. NORMAL ERROR PRINTOUTS
- B. EXTENDED ERROR PRINTOUTS

6.2.1 NORMAL ERROR PRINTOUTS

NORMAL ERROR PRINTOUTS ARE GENERATED BY THE "ERR" SUBROUTINE. THE ERR SUBROUTINE IS CALLED BY AN "ERROR" STATEMENT IN THE PROGRAM LISTING. THE NORMAL ERROR PRINTOUT TAKES THE FORM:

"PXX TYYY PC QZZZZZ ICNT VVVVV."

WHERE:

PXX IS THE NUMBER OF THE PROGRAM BEING RUN.
TTY Y Y Y IS THE NUMBER OF ROUTINE WHERE FAILURE OCCURRED.

PC QZZZZZ IS THE ADDRESS FROM WHICH THE ERROR CALLED WAS ISSUED.
ICNT VVVVV. IS NUMBER OF TIMES TEST WAS DONE WHEN FAILURE OCCURRED.
MEANINGFUL ONLY IN PRG0 AND PRG1.

AFTER THE PRINTOUT IS COMPLETED, THE PROGRAM WILL HALT AT COMMON ERROR HALT AT LOC 002516 IF SR15 IS SET.

WHEN THIS TYPE OF ERROR PRINTOUT OCCURS:

- A. IN THE PROGRAM LISTING, LOOK UP THE ADDRESS REFERENCED BY PCQZZZZZ.
- B. OPPOSITE THE PC VALUE AN ERROR STATEMENT WILL BE FOUND, AND IN THE COMMENTS SECTION A DESCRIPTION OF THE FAILURE WILL BE FOUND.
- C. AT THE BEGINNING OF THE TEST ROUTINE A DESCRIPTION OF THE TEST WILL BE FOUND.

6.2.2 EXTENDED ERROR PRINTOUTS

IN ADDITION TO THE INFORMATION TYPED BY THE NORMAL ERROR PRINTOUTS, THE EXTENDED ERROR PRINTOUTS TYPE INFORMATION THAT DESCRIBES THE TYPE OF FAILURE. MOST EXTENDED PRINTOUTS CONCERN THEMSELVES WITH DATA PROBLEMS. THE PRINTOUTS ARE GENERATED BY THE "ERRN" SUBROUTINE WHICH IS CALLED BY AN "ERRORN" STATEMENT IN THE PROGRAM LISTING. A TYPICAL PRINTOUT WOULD LOOK AS FOLLOWS:

"P05 T000 PC 011350 ICNT 00000. DATA ERROR S/B:0371 WAS:0071"

THE PROGRAM, TEST AND PC INFORMATION ARE THE SAME AS FOR NORMAL ERROR PRINTOUTS. THE PC VALUE ALTHOUGH HAVING THE SAME MEANING, IS NOT AS MEANINGFUL, SINCE THE ERRN SUBROUTINE MAY BE BEING CALLED BY A COMMON DATA ERROR SUBROUTINE WHICH IS USED BY MORE THAN ONE PROGRAM.

564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619

(6.2.2 CONT'D)

THE IMPORTANT INFORMATION IN AN EXTENDED ERROR PRINTOUT IS THE "EXTENDED" INFORMATION TYPED. SOME OF THE EXTENDED PRINTOUTS ARE DESCRIBED BELOW:

"DATA ERROR S/B XXXX WAS: YYYY"

DATA READ WITH READER DOES NOT AGREE WITH EXPECTED DATA. S/B XXXX (SHOULD BE) IS THE EXPECTED DATA. WAS YYYY IS THE RECEIVED DATA. DEPENDING ON THE PROGRAM, THE FAILURE COULD BE CAUSED BY THE READER OR THE PUNCH. EXAMINING THE TAPE WILL SHOW IF THE TAPE IS PUNCHED CORRECTLY.

"REREAD ERROR. 1ST READ: XXXX WAS: YYYY"

THIS ERROR PRINTOUT IS GENERATED BY PRGD TEST14. IT INDICATES THAT A REREAD OF THE READER BUFFER DID NOT AGREE WITH THE ORIGINAL DATA READ FROM THE BUFFER.

"SYNC ERROR"

THIS PRINTOUT INDICATES THAT A PROGRAM WAS UNSUCCESSFUL IN SYNCING UP WITH THE SPECIAL BINARY COUNT PATTERN TAPE IN THE READER, OR IN THE CASE OF PRG2, THAT THE PROGRAM HAS NOT READ A SUFFICIENT NUMBER OF ZEROES BEFORE SYNCING UP WITH THE LEADER CHARACTER (377). IF HALTED, PRESS CONTINUE TO TRY AGAIN.

"LEADER ERROR S/B: 377 WAS: XXXX" OR
"LEADER ERROR S/B BETWEEN 0 AND 3. WAS: XXXX"

ONE OR BOTH OF THESE PRINTOUTS IS GENERATED BY PRG2 WHEN IN READING THE LEADER THAT PRECEDES THE SPECIAL BINARY COUNT PUNCHED BY PRG3 THE DATA DOES NOT AGREE WITH THE EXPECTED DATA. CHECK THAT THE TAPE IS PUNCHED CORRECTLY. REFER TO PRG1 AND PRG2 DESCRIPTION.

"MATCH ERROR"

THIS PRINTOUT IS GENERATED BY PRG5 WHEN UNSUCCESSFUL IN MATCHING UP THE DATA READ FROM THE READER WITH THE EXPECTED DATA AS SPECIFIED BY SR. CHECK THAT THE TAPE IS THE ONE TO BE READ AND RESTART THE PROGRAM.

"FALSE READER INTERRUPT" OR,
"FALSE PUNCH INTERRUPT"

THE PROGRAM DID NOT FIND THE ERROR OR THE DONE BIT SET FOLLOWING AN INTERRUPT. POSSIBLY NOISE COULD BE CAUSING THE PROBLEM.

620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672

673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725

7. MISCELLANEOUS

7.1 TEST TAPES

THE FOLLOWING TEST TAPES ARE RELEASED WITH THIS PROGRAM:

- A. MAINDEC-00-D2G4-PT SPECIAL BINARY COUNT PATTERN TEST TAPE.
- B. MAINDEC-00-D2G2-PT ONES AND ZEROES TEST TAPE.

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.

7.2 SR OPTIONS

THE STANDARD SR OPTIONS ARE DESCRIBED HERE.

SR15 - HALT ON ERROR.

SR14 - SCOPE. THIS OPTION CAUSES THE PROGRAM TO REMAIN IN THE CURRENT TEST ROUTINE. WHEN THE OPTION IS REMOVED THE PROGRAM PERFORMS THE TEST THE NUMBER OF TIMES SPECIFIED BY ITS ITERATION COUNT, BEFORE GOING ON TO THE NEXT ROUTINE.

SR13 - INHIBIT ERROR PRINT. THIS OPTION IF SET WILL REMOVE ALL ERROR PRINTOUTS.

SR11 - INHIBIT ITERATION. SOME PROGRAMS CONSIST OF INDIVIDUAL TEST ROUTINES. FOR EACH ROUTINE THE FUNCTION BEING TESTED CAN BE TESTED A VARIABLE NUMBER OF TIMES BEFORE THE ROUTINE IS COMPLETED. THE NUMBER OF TIMES THE TEST IS TO BE PERFORMED IS CALLED THE ITERATION COUNT AND IT MAY DIFFER FROM ROUTINE TO ROUTINE. SETTING SR11 WILL CAUSE THE PROGRAM TO PERFORM ONLY ONE ITERATION FOR EACH ROUTINE DURING WHICH THE SWITCH IS SET. TWO POSSIBLE USES OF THIS OPTION ARE:

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

(7.2 CONT'D)

- A. QUICK PASS. WHEN A PROGRAM RUNS FOR SEVERAL MINUTES FOR ONE PROGRAM PASS, THE USER MAY ELECT TO RUN THROUGH THE PROGRAM QUICKLY TO FIND OUT IF ANY FAILURES SHOW UP 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 WITH A MULTIPLE ITERATION COUNT HAS DETECTED A SOLID FAILURE, THE ERROR WILL BE REPORTED MANY TIMES. TO GO ON TO THE NEXT ROUTINE IF DESIRED, THE USER CAN INHIBIT ITERATION. IT WILL BE NECESSARY TO SET SR11 ROUTINE AND HALT, TO CAUSE THE PROGRAM TO STOP AT END OF FAILING ROUTINE. OTHERWISE THE PROGRAM WILL QUICKLY RUN THROUGH THE NEXT ROUTINE ALSO.

SR10 - HALT AT END OF CURRENT ROUTINE. PRG0 AND PRG1 CONSIST OF INDIVIDUALLY NUMBERED TEST ROUTINES. SETTING SR10 WILL CAUSE PROGRAM TO HALT UPON COMPLETION OF CURRENT ROUTINE.

SR9 - SELECT ROUTINE. FOR PROGRAMS THAT CONSIST OF INDIVIDUAL TEST ROUTINES, THE USER MAY ELECT TO RUN ONLY A SPECIFIED ROUTINE. TO SELECT A ROUTINE SR9 MUST BE SET, AND SR7 THROUGH SR0 MUST BE SET TO THE NUMBER OF THE DESIRED ROUTINE. THE SELECTED NUMBER MUST BE A VALID ROUTINE NUMBER FOR THE PROGRAM BEING RUN, OR A USER ERROR PRINTOUT WILL OCCUR. THE PROGRAM WILL RUN THE SELECTED ROUTINE UNTIL THE SELECT ROUTINE OPTION IS CLEARED, OR UNTIL THE SELECTED ROUTINE NUMBER IS CHANGED. IF THE OPTION IS CLEARED, THE PROGRAM WILL PROCEED TO EXECUTE THE REMAINING ROUTINES IN THE PROGRAM. IF THE ROUTINE NUMBER IS CHANGED, THE PROGRAM WILL EXECUTE THE NEWLY SELECTED ROUTINE.

7.3 TESTING AT NON-STANDARD ADDRESSES AND/OR VECTORS

THIS PROGRAM CAN TEST PA611'S ASSIGNED TO NON-STANDARD ADDRESSES. ALL READERS MUST BE ASSIGNED CONSECUTIVE ADDRESSES, AND ALL PUNCHES MUST BE ASSIGNED CONSECUTIVE ADDRESSES.

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

LOCATION	FROM STANDARD	TO NON-STANDARD
001220	172600	1ST READER CSR ADDRESS
001222	172700	1ST PUNCH CSR ADDRESS

- B. IF THE TELETYPE IS ALSO AT NON STANDARD ADDRESSES, PERFORM THE FOLLOWING CHANGES:

LOCATION	FROM STANDARD	TO NON-STANDARD
001224	177560	TTY READER CSR ADDRESS
001226	177562	TTY READER BUFFER ADDRESS
001230	177564	TTY PRINTER CSR ADDRESS
001232	177566	TTY PRINTER BUFFER ADDRESS
001234	000060	TTY READER INTERRUPT VECTOR ADDRESS
001236	000200	TTY READER PRIORITY LEVEL
001240	000064	TTY PRINTER INTERRUPT VECTOR ADDRESS
001242	000200	TTY PRINTER PRIORITY LEVEL

- C. PROCEED TO USE THE PROGRAM, OR
- D. USING STANDARD DUMP ROUTINES, DUMP OUT THE ENTIRE PROGRAM IN ABSOLUTE FORMAT TO HAVE AN UPDATED OBJECT TAPE THAT REFLECTS YOUR SYSTEM, OR
- E. DUMP OUT ONLY LOCATIONS 001224 THROUGH 001242 IN ABSOLUTE FORMAT, 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 IS LOADED, IN ORDER TO OVERLAY LOCATIONS 001224 THROUGH 001242.

PAGE 14

8. DESCRIPTION
-----8.1 PRGO PROGRAM DESCRIPTION

PRGO IS THE PA611 READER TEST. IT CONSISTS OF 22 ROUTINES NUMBERED FROM 00 TO 24(8). THE PROGRAM USES A SPECIAL BINARY COUNT PATTERN TEST TAPE LOOP IN ALL ROUTINES.

ROUTINES 00 THROUGH 17 ARE BASIC LOGIC TESTS. ROUTINES 20 THROUGH 24 ARE READER EXERCISER TESTS. IN ROUTINES 20 THROUGH 24 THE READER WILL RESYNC ITSELF AFTER 3 DATA ERRORS HAVE OCCURRED.

763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818

819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
8628.2 PRG1 PROGRAM DESCRIPTION

PRG1 IS THE PA611 PUNCH TEST. IT CONSISTS OF 16 ROUTINES NUMBERED FROM 00 TO 17(8). ROUTINES 00 THROUGH 13 ARE BASIC LOGIC TESTS. ROUTINES 14 THROUGH 17 EXERCISE THE PUNCH USING THE FOLLOWING FORMAT:

- A. 20 BLANK CHARACTERS
- B. SYNC CHARACTER RUBOUT.
- C. MODE NUMBER (BETWEEN 0 AND 3)
- D. 4 BLANK CHARACTERS
- E. 512 CHARACTERS OF SPECIAL BINARY COUNT PATTERN.

RTN14 - PUNCHES 5 DATA BLOCKS AT FULL SPEED.

RTN15 - PUNCHES 5 DATA BLOCKS. THE SPECIAL BINARY COUNT PATTERN DATA IS PUNCHED WITH RANDOM STALLS OF UP TO 47 MSECS. AFTER EACH CHARACTER.

RTN16 - PUNCHES 5 DATA BLOCKS. THE SPECIAL BINARY COUNT PATTERN DATA IS PUNCHED WITH RANDOM STALLS OF UP TO 47 MILLISECONDS BETWEEN GROUPS OF CHARACTERS OF UP TO 15 CHARACTERS.

RTN17 - PUNCHES 1 DATA BLOCK. THE SPECIAL BINARY COUNT PATTERN DATA IS PUNCHED WITH A 5 SECOND STALL PRECEDING EACH 32 CHARACTER GROUP PUNCHED.

8.3 PRG2 PROGRAM DESCRIPTION

PRG2 VERIFIES THE PAPER TAPE PRODUCED BY PRG1. THE PROGRAM CONSISTS OF A SINGLE ROUTINE THAT PERFORMS THE FOLLOWING STEPS:

- A. LOOK FOR 10 CONSECUTIVE 0 CHARACTERS
- B. LOOK FOR SYNC CHARACTER (RUBOUT)
- C. LOOK FOR MODE #. BETWEEN 0 AND 3.
- D. READ 4 BLANK CHARACTERS
- E. READ 512 BINARY CHARACTERS.
- F. GO TO STEP A.

THE ROUTINE WILL REPORT EVERY ERROR. IT WILL NOT RESYNC ON THE SPECIAL BINARY COUNT PATTERN, SINCE IT IS INTENDED THAT EVERY ERROR CAUSED BY THE PUNCH BE REPORTED.

863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
9078.4 PRG3 COMBINED READER-PUNCH TEST

THIS CONTINUOUS RUNNING PROGRAM EXERCISES THE PUNCH AND READER CONCURRENTLY. THE SPECIAL BINARY COUNT PATTERN IS USED IN THIS PROGRAM,

- A. THE PUNCH PUNCHES DATA AT FULL SPEED. WHEN THE CHARACTER COUNT REACHES 20, THE PUNCH ROUTINE ENABLES THE READER.
- B. WHEN THE CHARACTER COUNT REACHES 40, THE PUNCH ROUTINE WILL STOP PUNCHING. PUNCHING WILL NOT RESUME UNTIL THE CHARACTER COUNT IS DECREMENTED TO 31 BY THE READ ROUTINE.
- C. IF THE CHARACTER COUNT IS OVER 31, THE READER READS AT FULL SPEED.
- D. IF THE CHARACTER COUNT IS 31 OR LESS THE READER WILL READ WITH RANDOM STALLS BETWEEN CHARACTERS.
- E. IF THE CHARACTER COUNT BECOMES 0, THE READER STOPS READING UNTIL THE COUNT CLIMBS TO 20.
- F. THE READ ROUTINE WILL RESYNC AUTOMATICALLY AFTER 3 ERRORS.

8.5 PRG4 PROGRAM DESCRIPTION

PRG4 WILL PUNCH CONTINUOUSLY THE 2 CHARACTERS WHOSE CODES ARE SET IN THE SR. THE ROUTINE IS USED FOR GENERATING ALL 0'S TAPE, ALL 1'S TAPE, ONES AND ZEROES TAPE, ETC.

8.6 PRG5 PROGRAM DESCRIPTION

PRG5 READS AND CHECKS A TAPE PUNCHED WITH THE CHARACTERS WHOSE CODES ARE SET IN THE SR. THIS ROUTINE IS USEFUL IN SETTING UP THE READ PHOTOCELLS AND READ AMPLIFIER.

8.7 PRG6 PROGRAM DESCRIPTION

PRG6 WILL ENABLE THE READER FOR THE NUMBER OF CHARACTERS SPECIFIED IN THE LEFT HALF OF THE SR, AND THEN IT WILL STALL FOR THE NUMBER OF MILLISECONDS SPECIFIED IN THE RIGHT HALF OF THE SR. THIS ROUTINE IS USEFUL IN SETTING UP THE READER CLOCK, ACCELERATOR, STROBE, AND FOR CHECKING THE STOP DELAY.

908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
9638.8 PRG7 PROGRAM DESCRIPTION

PRG7 PUNCHES THE SPECIAL BINARY COUNT PATTERN CONTINUOUSLY.

8.9 PRG10 PROGRAM DESCRIPTION

PRG10 IS A ROUTINE USED TO CHECK THE SPEED OF THE READER.
READER SPEED CAN BE MEASURED IN TWO WAYS:

- A. COARSE. 30 SECOND TIMING. PLUS OR MINUS 10 CHARACTER ACCURACY.
- B. FINE. 300 SECOND TIMING. PLUS OR MINUS 1 CHARACTER ACCURACY.

SR14 INDICATES TO THE ROUTINE THE TIMING PERIOD THE USER IS
GOING TO USE. SR14=0 INDICATES 30 SECOND TIMING.

THE USER CONTROLS THE DURATION OF THE TIMING PERIOD BY USING A
SWEEP SECOND HAND WATCH OR STOP-WATCH. AT THE END OF THE
TIMING PERIOD, SR15 IS SET TO A 1 TO OBTAIN A SPEED PRINTOUT.

8.10 PRG11 PROGRAM DESCRIPTION

PRG11 IS USED TO CHECK THE SPEED OF THE PUNCH. THE ROUTINE
USES A 60 SECOND TIMING PERIOD THAT IS CONTROLLED BY THE USER.
AT THE END OF THE TIMING PERIOD SR15 IS SET TO A 1 TO OBTAIN A
SPEED PRINTOUT.

8.11 PRG12 PROGRAM DESCRIPTION

PROGRAM 12 IS USED TO TEST THE PROGRAMMABLE INIT FEATURE
ADDED TO THE PUNCH CONTROL LOGIC. THE PROGRAM PUNCHES AN
ENDLESS BINARY PATTERN, WAITING FOR THE OPERATOR TO HANG THE
PUNCH UP. THIS CAN USUALLY BE ACCOMPLISHED BY TURNING THE
CONTROL SWITCH OFF AND JIGGLING THE TAPE LOW SENSOR SWITCH
WHILE THE PUNCH IS PUNCHING. IF THE PUNCH CYCLE HAS NOT BEEN
COMPLETED, READY WILL REMAIN LOW AND THE PUNCH WILL HANG.

THE PROGRAM SENSES THIS CONDITION IN A WATCHDOG TIMER LOOP.
WHEN THE PROGRAM SEES THAT THE PUNCH IS HUNG, THEN IT
TIMES OUT AND HALTS.

WHEN THE OPERATOR HITS THE CONTINUE SWITCH, AN INIT PULSE IS
GENERATED BY THE PROGRAM TO RE-INIT THE PUNCH. [THIS INIT
PULSE DOES NOT EFFECT OTHER DEVICES ON THE BUS.]

UPON HITTING THE CONTINUE SWITCH,
THE PROCESS REPEATS ITSELF.

8.12 PRG13 PROGRAM DESCRIPTION

964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982

THIS PROGRAM IS USED TO CHECK THE ABILITY OF THE READER
LIGHT TO BE TURNED OFF UNDER PROGRAM CONTROL.

ASSUMING THAT THE READER LIGHT IS ON, THE STEPS PERFORMED
BY THE PROGRAM ARE:

1. GET THE ADDRESS OF THE READER TO BE TESTED (READER
IS SELECTED VIA CONSOLE TTY RESPONSE)
2. CHECK SWITCH 8. IF SWITCH 8=1 THEN ISSUE A COMMAND TO
TURN THE LIGHT OFF. IF SWITCH 8=0 THEN RECHECK THE
SWITCH AND LOOP
ON THIS STEP.
3. CHECK SWITCH 12.
IF SW12=1 THEN GO TO STEP 1.
IF SW12=0 THEN GO TO STEP 2.

983
984
985
986
987
988
989
990
991

```

%
;STORED IN DECTAPE LIB12 - MIKE MITCHELL 3/75
;TYPESET 11 READER-PUNCH TESTS
;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
  .TITLE PA611
  .ABS
  .ENABLE AMA
  .LIST ME
  .NLIST MD,SEQ
;PRG0 - READER TEST
;PRG1 - PUNCH TEST
;PRG2 - PUNCH VERIFY ROUTINE
;PRG3 - COMBINED READER-PUNCH TEST
;PRG4 - PUNCH 2 CHARACTERS FROM SR.
;PRG5 - READ 2 CHARACTERS AS PER SR.
;PRG6 - READ X CHARS, STALL Y MSECS.
;PRG7 - PUNCH SPECIAL BINARY COUNT PATTERN TAPE.
;PRG10 - READER SPEED PRINT ROUTINE.
;PRG11 - PUNCH SPEED PRINT ROUTINE.
;PRG12 - PUNCH INIT TEST
;PRG13 - READER LIGHT TEST.

;SR 15 - HALT-ON-ERROR.
;SR 14 - SCOPE.
;SR 13 - INHIBIT ERROR PRINT.
;SR 11 - INHIBIT ITERATION.
;SR 10 - HALT AT END OF CURRENT ROUTINE.
;SR 9 - SELECT ROUTINE.
;SR 8 - TURN READER LIGHT OFF (PRG13 ONLY)
;SR 7 THROUGH SR 0 - NUMBER OF ROUTINE TO BE SELECTED.

;PA611 ADDRESSES:
;PUNCH.....772776-772700
;READER.....772676-772600
;RDR0-----172600
;RDR1-----172604
;RDR2-----172610
;RDR3-----172614

```

```

000000 000000
000000 000002
000002 000000
000004 000006
000006 000000
000010 000012
000012 000000
000014 002336
000016 000340
000020 002366
000022 000340
000024 000026
000026 000000
000030 002136

```

```

;
;.=0
;+2
MACHER: HALT ;UNASSIGNED TRAP
;+2
;HALT ;SP OVERFLOW, BUS ERROR TRAP
;+2
;HALT ;RESERVED INSTRUCTION TRAP
TRCV: SV5S ;TRACE TRAP
;PRTY7
IOTV: RS5S ;TRAP TO CALL IOX
;PRTY7
;+2
;HALT ;POWER FAIL TRAP
EMTV: EMTINT ;EMT TRAP

```

000032 000340
000034 000036
000036 000002
000040

TRPV: PRTY7
. +2 ; TRAP TRAP. SIMILAR TO EMT.
RTI ; EXIT TRAP CALL.
. =40

; LOCATIONS 40 THROUGH 776 ARE FILLED WITH .+2 AND HALT.

; EQUATE STATEMENTS

177570
177776
001200
000240
000000
100000
100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001
000000
000001
000002
000003
000004
000005
000006
000007
000007
005746
024646
005726
022626
000340
000300
000240
000200
000140
000100
000040
000000
000007
177777
177777
177777
000003

SR=177570
PSW=177776
SPBOT=1200
NOP=240
OPEN=0
MANUAL=BIT15
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
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
R6=%6
R7=%7
PC=%7
PUSH=005746
PUSH2=024646
POPSP=005726
POPSP2=022626
PRTY7=340
PRTY6=300
PRTY5=240
PRTY4=200
PRTY3=140
PRTY2=100
PRTY1=40
PRTY0=0
BELL=007
POTLST=-1
PITLST=-1
TLAST=-1
TRC=3

	000040		I=40	
	100000		A=BIT15	
	040000		B=BIT14	
	020000		C=BIT13	
	000100		IE=BIT6	
	000000		EMTX=0	
	000003		SAVSS=3	
	000004		RSTSS=4	
	000060		=60	
000060	002450		KBSVC	;KEYBOARD VECTOR SETUP.
000062	000340		PRTY7	;STATUS UPON KEYBOARD INTERRUPT.
	000200		=200	
000200	000137	001450	JMP	START
	001000		=1000	
	000240		NOP	
001002	000240		NOP	
001004	005037	001252	CLR	INITD
001010	000137	001450	JMP	START
	001200		=1200	
001200	000000		PRS: OPEN	;READER CSR
001202	000000		PRB: OPEN	;READER BUFFER
001204	000000		PPS: OPEN	;PUNCH CSR
001206	000000		PPB: OPEN	;PUNCH BUFFER
001210	000000		RDRVTR: OPEN	;READER INTERRUPT VECTOR
001212	000200		RDLVL: PRTY4	;READER PRIORITY LEVEL
001214	000000		PCHVTR: OPEN	;PUNCH INTERRUPT VECTOR
001216	000200		PCHVL: PRTY4	;PUNCH PRIORITY LEVEL
001220	172600		FSTRDR: 172600	;ADDR OF 1ST READER.
001222	172700		FSTPCH: 172700	;ADDR OF 1ST PUNCH.
001224	177560		TKS: 177560	;LSR CSR
001226	177562		TKB: 177562	;LSR BUFFER
001230	177564		TPS: 177564	;LSP CSR
001232	177566		TPB: 177566	;LSP BUFFER
001234	000060		TKVTR: 60	;LSR INTERRUPT VECTOR
001236	000200		TKLVL: PRTY4	;LSR PRIORITY LEVEL
001240	000064		TPVTR: 64	;LSP INTERRUPT VECTOR
001242	000200		TPLVL: PRTY4	;LSP PRIORITY LEVEL
001244	000000		FSTVCT: OPEN	
001246	000000		RDLIM: OPEN	
001250	000000		PCHIM: OPEN	
001252	000000		INITD: OPEN	
001254	000000		PRGNUM: OPEN	;CONTAINS CURRENT PROGRAM#
001256	000000		KSTART: OPEN	;CURRENT PROGRAM START ADDRESS.
001260	000000		CURTST: OPEN	;CONTAINS ADDR OF CURRENT TEST.
001262	000000		RTNNO: OPEN	;CONTAINS CURRENT TEST #.
001264	000000		NXTST: OPEN	;CONTAINS ADDR OF NEXT TEST.
001266	000000		ICNT: OPEN	
001270	000000		ICTR: OPEN	;CONTAINS CURRENT ITERATION COUNT
001272	000000		SCOPTR: OPEN	;CONTAINS CURRENT SCOPE POINTER.
001274	000000		PRGID: OPEN	;CONTAINS PROGRAM INDICATORS
001276	005204		PRGTAB: PRG0	;PRG0 START ADDRESS
001300	006656			;PRG1 START ADDRESS
001302	010146			;PRG2 START ADDRESS
001304	010530			;PRG3 START ADDRESS
001306	011302			;PRG4 START ADDRESS
001310	011354			;PRG5 START ADDRESS

001312 011634
001314 011722
001316 011774
001320 012116

PRG6
PRG7
PRG10
PRG11

;PRG6 START ADDRESS
;PRG7 START ADDRESS
;PRG10 START ADDRESS
;PRG11 START ADDRESS

001322 012246
001324 012502

;OPERATOR INTERVENTION TESTS:

PRG12
PRG13

;PUNCH TEST (OPR INTV.)
;READER LIGH TEST
;(OPR INTV REQD.)

001326 000000
001330 000000
001332 000000
001334 000000
001336 000000
001340 000000
001342 000000
001344 000000
001346 000000
001350 000000
001352 000000
001354 000000
001356 000000
001360 000000
001362 000000

ERRT: OPEN
RCNT: OPEN
PCHOUT: OPEN
CRBUF: OPEN
CHR1: OPEN
CHR2: OPEN
CHR3: OPEN
ERCTR: OPEN
PCHMSK: OPEN
RDRMSK: OPEN
CTRA: OPEN
CTRB: OPEN
CTRC: OPEN
CTRD: OPEN
FPC: OPEN

;CHARACTER COUNT
;HOLDS ONE CHARACTER FROM READER.

.NLIST MC

001364 001752
001364 002412
001366 002156
001370 002256
001372 002206
001374 002306
001376 002176
001400 002302
001402 002702
001404 002774
001406 004152
001410 003744
001412 003774
001414 002526
001416 002536
001420 003304
001422 003376
001424 003560
001426 003636
001430 003354
001432 002502
001434 002514
001436 003070
001440 003234
001442 003156
001444 003024

EMTTAB: CHAINN
SRSETT
SV03
RS03
SV05
RS05
SV05S
RS05S
TYP
TYP5
STAL
STPTRV
STPTPV
ERR
ERN
OACNVV
BDCNVV
RNGEN
INRNDN
BMOVV
CHLT
EHLT
INBINN
GTBINR
GTBINP
DLYN

;POINTER FOR EMT CALL SCOPE
;POINTER FOR EMT CALL SRESET
;POINTER FOR EMT CALL SAV03
;POINTER FOR EMT CALL RST03
;POINTER FOR EMT CALL SAV05
;POINTER FOR EMT CALL RST05
;POINTER FOR EMT CALL SAV05S
;POINTER FOR EMT CALL RST05S
;POINTER FOR EMT CALL TYPE
;POINTER FOR EMT CALL TYPES
;POINTER FOR EMT CALL STALL
;POINTER FOR EMT CALL STRDRV
;POINTER FOR EMT CALL STPCHV
;POINTER FOR EMT CALL ERROR
;POINTER FOR EMT CALL ERRORN
;POINTER FOR EMT CALL OACNV
;POINTER FOR EMT CALL BDCNV
;POINTER FOR EMT CALL RNDNUM
;POINTER FOR EMT CALL INRND
;POINTER FOR EMT CALL BMOVE
;POINTER FOR EMT CALL CHALT
;POINTER FOR EMT CALL EHALT
;POINTER FOR EMT CALL INBIN
;POINTER FOR EMT CALL GETBNR
;POINTER FOR EMT CALL GETBNP
;POINTER FOR EMT CALL DELAY

001450 012706 001200
001454 005737 001252
001460 001041
001462 !04010

START: MOV #SPBOT,R6
TST INITD
BNE STRTA
TYPE

;SET BOTTOM OF SP STACK.
;SEE IF PROGRAM IS INITIALIZED.
;BR IF YES.
;TYPE TITLE.

001464	012712				PGTIT			
001466	104011				TYPES			;TYPE INSTRUCTIONS TO SET RDR 0 VECTOR.
001470	012775				MSVCTR			
001472	014001				IM23			
001474	177777				-1			
001476	104024				CHALT			
001500	013737	177570	001244		MOV	SR,FSTVCT		;SAVE RDR0 VECTOR.
001506	104011				TYPES			;TYPE INSTRUCTIONS TO SET # OF READERS.
001510	013027				SELRDR			
001512	014001				IM23			
001514	177777				-1			
001516	104024				CHALT			
001520	113737	177570	001246		MOVB	SR,RDR LIM		;SAVE # OF READERS.
001526	104011				TYPES			;TYPE INSTRUCTIONS TO SET # OF PUNCHES.
001530	013060				SELPCH			
001532	014001				IM23			
001534	177777				-1			
001536	104024				CHALT			
001540	113737	177570	001250		MOVB	SR,PCH LIM		;SAVE # OF PUNCHES.
001546	012737	177777	001252		MOV	#-1,INITD		;INITIALIZATION COMPLETE.
001554	104010				TYPE			;RESTART MESSAGE.
001556	012753				RUNINS			
001560	104024				CHALT			
001562	000776				BR	.-2		
001564	012737	000340	177776	STRTA:	MOV	#PRTY7,PSW		;SET PRIORITY 7.
001572	004737	005134			JSR	PC,CLNUP		;GO DO CLEAN-UP.
001576	013700	177570			MOV	SR,RO		;GET PROGRAM NUMBER.
001602	042700	177760			BIC	#177760,RO		
001606	020027	000013			CMP	RO,#13		;VALID PROGRAM NUMBER?
001612	101404				BLOS	STRTB		;BR IF YES.
001614	104010				TYPE			;TYPE INCORRECT PROGRAM MESSAGE.
001616	012601				CM2			
001620	104024				CHALT			
001622	000760				BR	STRTA		;TRY AGAIN.
001624	010037	001254		STRTB:	MOV	RO,PRGNUM		;SAVE PROGRAM NUMBER.
001630	006300				ASL	RO		;RO TIMES 2.
001632	104001				SRESET			;SYSTEM RESET.
001634	000170	001276		SRSET:	JMP	@PRGTAB(0)		;GO TO SELECTED PROGRAM.
001640	104011				TYPES			;TYPE SR OPTION MESSAGE.
001642	012644				ASETSR			
001644	014001				IM23			
001646	177777				-1			
001650	104024				CHALT			;COMMON HALT.
001652	013737	001256	001264	GETRDY:	MOV	KSTART,NXTST		;ADDR OF 1ST ROUTINE TO NXTST
001660	012737	000340	177776	GTRDYX:	MOV	#PRTY7,PSW		;SET PRIORITY 7.
001666	012706	001200			MOV	#SPBOT,R6		;SET BOTTOM OF STACK.
001672	104001				SRESET			;ISSUE RESET.
001674	004737	002076		GTRDYA:	JSR	R7,FRWD		;ROLL FORWARD TO "NEXT" ROUTINE.
001700	032737	001000	177570	GTRDYB:	BIT	#BIT9,SR		;CHECK SELECT ROUTINE SWITCH
001706	001002				BNE	GTRDYC		;BRANCH IF SELECT ROUTINE SWITCH IS SET.
001710	000177	177344		GORUN:	JMP	@CURTST		;GO RUN CURRENT ROUTINE.
001714	013700	177570		GTRDYC:	MOV	SR,RO		; (SR) TO RO
001720	042700	177600			BIC	#177600,RO		;MASK UNDESIRED BITS
001724	123700	001262			CMPB	RTNNO,RO		;COMPARE RTNNO TO (RO)
001730	001767				BEQ	GORUN		;BR IF ROUTINE FOUND.
001732	022737	177777	001264	GTRDYD:	CMP	#-1,NXTST		;NO. CHECK FOR LAST ROUTINE.

001740	001355			BNE	GTRDYA				;BRANCH IF NOT LAST ROUTINE.
001742	104010			TYPE					;TYPE INCORRECT RTN SELECTED.
001744	012624			CM3					
001746	104024			CHALT					;COMMON HALT.
001750	000740			BR	GETRDY				;START OVER.
001752	012706	001200		CHAINN: MOV	#SPBOT,R6				;RESTORE STACK.
001756	005237	001266		INC	ICNT				;INCREMENT ITERATION COUNT.
001762	001002			BNE	CHNAC				;BR IF RESULT NOT 0.
001764	005137	001266		COM	ICNT				;RESULT 0. RESET ICNT TO -1.
001770	032737	040000	177570	CHNAC: BIT	#BIT14,SR				;CHECK FOR SCOPE OPTION.
001776	001402			BEQ	CHNA				;BRANCH IF SCOPE SW NOT SET.
002000	000177	177266		CHNAB: JMP	SCOPTTR				;RETURN TO ROUTINE.
002004	032737	004000	177570	CHNA: BIT	#BIT11,SR				;TEST INHIBIT ITERATION SWITCH
002012	001003			BNE	CHNAA				;BRANCH IF INHIBIT ITERATION SW SET.
002014	005337	001270		DEC	ICTR				;DECREMENT ITERATION COUNT.
002020	001367			BNE	CHNAB				;BRANCH IF COUNT NOT 0.
002022	032737	002000	177570	CHNAA: BIT	#BIT10,SR				;ROUTINE END HALT SW SET? (SR10)
002030	001403			BEQ	CHNB				;BRANCH IF NOT SET.
002032	013700	001262		MOV	RTNNO,RO				;TEST # TO RO.
002036	000000			HALT					;ROUTINE END HALT. TEST # IN LIGHTS.
002040	032737	001000	177570	CHNB: BIT	#BIT9,SR				;CHECK SELECT ROUTINE SWITCH
002046	001301			BNE	GETRDY				;BRANCH IF SELECT RTN SW SET
002050	022737	177777	001264	CMP	#-1,NXTST				;LAST TEST?
002056	001300			BNE	GTRDYX				;BRANCH IF NOT LAST TEST.
002060	104017			OACNV					;CONVERT PROGRAM NUMBER TO ASCII.
002062	001254			PRGNUM					
002064	012571			APN					
002066	000002			2					
002070	104010			TYPE					;TYPE PROGRAM END BELL.
002072	012566			APGEND					
002074	000666			BR	GETRDY				;GO REPEAT PROGRAM.
002076	013705	001264		FORWD: MOV	NXTST,R5				;ADDR OF NEXT ROUTINE TO R5.
002102	012537	001262		MOV	(5)+,RTNNO				;GET NEXT ROUTINE NUMBER.
002106	012537	001264		MOV	(5)+,NXTST				;GET ADDR OF NEXT "NEXT" ROUTINE.
002112	012537	001270		MOV	(5)+,ICTR				;GET ITERATION COUNT.
002116	012537	001272		MOV	(5)+,SCOPTTR				;GET SCOPE LOOP ENTRY POINTER.
002122	010537	001260		FORWDA: MOV	R5,CURTST				;ADDR OF NOW CURRENT TEST TO CURTST.
002126	012737	000001	001266	MOV	#1,ICNT				;PRESET ICNT TO 1.
002134	000207			RTS	R7				;EXIT FORWD SUBROUTINE.
				;EMT INTERPRETER ROUTINE.					
002136	010046			EMTINT: MOV	RO,-(6)				;PUSH RO.
002140	016600	000002		MOV	2(6),RO				;GET EMT PC.
002144	014000			MOV	-(C),RO				;GET EMT CALL.
002146	006300			ASL	RO				;TIMES 2.
002150	016000	171364		MOV	EMTTAB-10000(0),RO				;FORM EMT ROUTINE ADDR.
002154	000200			RTS	RO				;GO TO EMT ROUTINE; RESTORE RO.
				;SAVE REGS 0 TO 3 SUBROUTINE.					
002156	012666	177766		SV03: MOV	(6)+,-10.(6)				;MOVE PC UPSTACK.
002162	012666	177766		MOV	(6)+,-10.(6)				;MOVE STATUS UPSTACK.
002166	012737	000002	002242	MOV	#RTI,SV05C				
002174	000415			BR	SV05B				
				;SUB TO SAVE REGS 0 TO 5 AND PLACE EMT PC IN R5.					
002176	012737	000240	002242	SV05S: MOV	#NOP,SV05C				
002204	000403			BR	SV05A				
				;SUB TO SAVE REGS 0 TO 5.					
002206	012737	000002	002242	SV05: MOV	#RTI,SV05C				

F03

PA611 MACY11 27(1006) 04-NOV-76 12:14 PAGE 31
 DZPAAA.CMB 04-NOV-76 12:11

002214	012666	177762	SV05A:	MOV (6)+,-14.(6)	;MOVE PC AND PSW UPSTACK.
002220	012666	177762		MOV (6)+,-14.(6)	
002224	010546			MOV R5,-(6)	
002226	010446			MOV R4,-(6)	
002230	010346		SV05B:	MOV R3,-(6)	
002232	010246			MOV R2,-(6)	
002234	010146			MOV R1,-(6)	
002236	010046			MOV R0,-(6)	
002240	024646			PUSH2	
002242	000002		SV05C:	RTI	;RTI OR NOP.
002244	016605	000020		MOV 16.(6),R5	;EMT PC TO R5.
002250	010504			MOV R5,R4	
002252	005744			TST -(4)	
002254	000002			RTI	;EXIT.
				;RESTORE REGS 0 TO 3 SUBROUTINE.	
002256	022626		RS03:	POPSP2	
002260	012600			MOV (6)+,R0	;RESTORE REGS 0 TO 4.
002262	012601			MOV (6)+,R1	
002264	012602			MOV (6)+,R2	
002266	012603			MOV (6)+,R3	
002270	016646	177766		MOV -10.(6),-(6)	;MOVE PC AND PSW DOWN STACK.
002274	016646	177766		MOV -10.(6),-(6)	
002300	000002			RTI	;EXIT.
				;SUB TO SET R5 IN EMT PC AND RESTORE REGS 0 TO 5.	
002302	010566	000020	RS05S:	MOV R5,16.(6)	;SET EMT PC TO R5 CONTENTS.
				;SUB TO RESTORE REGS 0 TO 5.	
002306	022626		RS05:	POPSP2	
002310	012600			MOV (6)+,R0	
002312	012601			MOV (6)+,R1	
002314	012602			MOV (6)+,R2	
002316	012603			MOV (6)+,R3	
002320	012604			MOV (6)+,R4	
002322	012605			MOV (6)+,R5	
002324	016646	177762		MOV -14.(6),-(6)	;MOVE PC AND PSW DOWNSTACK.
002330	016646	177762		MOV -14.(6),-(6)	
002334	000002			RTI	;EXIT.
002336	012666	177772	SV55:	MOV (6)+,-6(6)	;PC AND PSW UPSTACK.
002342	012666	177772		MOV (6)+,-6(6)	
002346	010546			MOV R5,-(6)	;SAVE R5.
002350	010446			MOV R4,-(6)	;SAVE R4.
002352	024646			PUSH2	
002354	016605	000010		MOV 8.(6),R5	;EMT PC TO R5.
002360	010504			MOV R5,R4	;EMT PC TO R4.
002362	005744			TST -(4)	
002364	000002			RTI	;EXIT EMT SUB.
002366	010566	000010	RS55:	MOV R5,8.(6)	;R5 TO EMT PC.
002372	022626			POPSP2	
002374	012604			MOV (6)+,R4	;RESTORE R4.
002376	012605			MOV (6)+,R5	;RESTORE R5.
002400	016646	177772		MOV -6(6),-(6)	
002404	016646	177772		MOV -6(6),-(6)	
002410	000002			RTI	;EXIT.
				;ROUTINE TO ISSUE RESET AND ENABLE KEYBOARD INTERRUPTS.	
002412	104004		SR5ETT:	SAV05	
002414	012700	052525		MOV #52525,R0	;DATA TO R0.
002420	005100			COM R0	;COMPLEMENT (R0).

002422	010037	002416			MOV	RD,SRSETT+4		;(RD) TO SRSETT+4.
002426	000005				RESET			;RESET. RD IS DISPLAYED.
002430	104005				RSTOS			
002432	005737	000042			TST	2#42		;LOADED FROM DECTAPE?
002436	001403				BEQ	SRSETA		;BR IF NOT.
002440	052777	000100	176556		BIS	#BIT6,2TKS		;ENABLE KEYBOARD INTERRUPTS.
002446	000002				RTI			;EXIT.
					SRSETA:	RTI		
								;KEYBOARD SERVICE ROUTINE.
					KBSVC:	MOV	2TKB,#0	;READ KEYBOARD BUFFER.
002450	017727	176552	000000					;CLEAR PARITY BIT.
002456	042737	000200	002454		BIC	#BIT7,KBSVC+4		;IS IT CTRL C?
002464	022737	000003	002454		CMP	#3,KBSVC+4		;BR IF YES.
002472	001401				BEQ	.+4		;NO. EXIT.
002474	000002				RTI			;EXIT TO DECTAPE MONITOR.
002476	013707	000042			MOV	2#42,PC		
								;COMMON HALT ROUTINE
					CHLT:	SAVOSS		
002502	104006							;DEVELOP ADDR OF CALLER.
002504	010400				MOV	R4,RD		;HALT CALL ADDR IN DATA LIGTHS.
002506	000000				HALT			
002510	104007				RSTOSS			
002512	000002				RTI			;EXIT.
								;CONDITIONAL ERROR HALT ROUTINE.
002514	005737	177570			EHLT:	TST	SR	;CHECK FOR HALT ON ERROR.
002520	100001					BPL	EHLTA	;BRANCH IF NO HALT DESIRED.
002522	000000					HALT		;HALT.
002524	000002					RTI		;IN DATA LIGHTS.
002526	012737	000406	002656		EHLTA:	RTI		;SET UP FOR SINGLE MESSAGE.
002534	000403				ERR:	MOV	#406,ERRNB	
002536	012737	000240	002656			BR	ERRNB+6	
002544	010437	001362			ERRN:	MOV	#NOP,ERRNB	;SET UP FOR MULTIPLE MESSAGES.
002550	104017					MOV	R4,FPC	;CONVERT CALL ADDR OF SUB CALLING.
002552	001362					OACNV		
002554	015032					FPC		
002556	000006					AFPC		
002560	000003					6		
002562	010537	002654				SAVSS		;SAVE REG 55
002566	162737	000002	002654			MOV	R5,ERRB	;DETERMINE CALLING ADDR.
002574	104017					SUB	#2,ERRB	
002576	002654					OACNV		;CONVERT CALLING ADDR TO ASCII.
002600	015000					ERRB		
002602	000006					APC		
002604	104017					6		;CONVERT PROGRAM # TO ASCII.
002606	001254					OACNV		
002610	014764					PRGNUM		
002612	000002					APNUMB		
002614	104017					2		;CONVERT TEST # TO ASCII.
002616	001262					OACNV		
002620	014771					RTNNO		
002622	000003					ATNUMB		
002624	104020					3		;CONVERT ICNT TO DECIMAL ASCII.
002626	001266					BDCNV		
002630	015015					ICNT		
002632	000005					AICNT		
002634	012737	014761	002654			5		;TYPE ERR HEADER MSG IF NOT INHIBITED.
002642	032737	020000	177570		ERRNA:	MOV	#EMO,ERRB	;INHIBIT ERR PRINT?
002650	001002					BIT	#BIT13,SR	;BR TO INHIBIT.
002652	104010					BNE	ERRNB	;TYPE MSG.
						TYPE		

```

002654 000000 ERRB: OPEN ; DESIRED MSG ADDR GOES HERE.
002656 000000 ERRNB: OPEN ; NOP OR 406
002660 012537 002654 MOV (5)+,ERRB ; GET ADDR OF NEXT MSG.
002664 022737 177777 002654 CMP #-1,ERRB ; TERMINATOR?
002672 001363 BNE ERRNA ; GO TYPE IF NOT TERMINATOR.
002674 104025 ERRNC: EHALT ; END OF MSGS. HALT IF REQUIRED.
002676 000004 RSTSS ; RESTORE REG 55.
002700 000002 RTI ; EXIT EMT SUB.

; SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
002702 104006 TYP: SAVO55
002704 012500 MOV (5)+,RO ; ADDRESS OF MESSAGE TO RO.
002706 112001 TYPA: MOVB (0)+,R1 ; GET CHARACTER
002710 122701 000100 CMPB #100,R1 ; CHECK FOR "Q" CHARACTER
002714 001002 BNE TYPC ; BRANCH IF NOT "Q".
002716 104007 RSTOSS
002720 000002 RTI ; TERMINATOR CHAR. DONE. EXIT.
002722 122701 000045 TYPC: CMPB #45,R1 ; CHECK FOR "%".
002726 001411 BEQ TYPF ; BRANCH IF "%".
002730 004737 002736 JSR R7,TYPD ; TYPE CHAR IN R1
002734 000764 BR TYPA
002736 110177 176270 TYPD: MOVB R1,@TPB ; OUTPUT CHARACTER TO PRINTER
002742 105777 176262 TSTB @TPS ; WAIT FOR DONE FLAG.
002746 100375 BPL -4
002750 000207 RTS R7 ; EXIT
002752 112701 000015 TYPF: MOVB #15,R1 ; MOVE CARRIAGE RETURN CODE TO R1
002756 004737 002736 JSR R7,TYPD ; GO TYPE CHAR.
002762 112701 000012 TYPG: MOVB #12,R1 ; MOVE LF CODE TO R1.
002766 004737 002736 JSR R7,TYPD ; GO TYPE CHAR.
002772 000745 BR TYPA

; SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
002774 000003 TYP5: SAV55
002776 012537 003020 MOV (5)+,TYP5B ; ADDR OF MESSAGE TO TYP5B.
003002 022737 177777 003020 CMP #-1,TYP5B ; CHECK FOR TERMINATOR
003010 001002 BNE TYP5A ; BRANCH IF NOT TERMINATOR.
003012 000004 RSTSS
003014 000002 RTI ; TERMINATOR, EXIT
003016 104010 TYP5A: TYPE ; CALL ON TYP SUB TO TYPE MESSAGE
003020 000000 TYP5B: OPEN ; ADDRESS OF MESSAGE GOES HERE
003022 000765 BR TYP5+2 ; GO PROCESS NEXT MESSAGE

; SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
003024 011637 003064 DLYN: MOV (6),DLCNT ; GET EMT PC.
003030 062716 000002 ADD #2,(6) ; SET UP EXIT PC.
003034 104002 SAVO3 ; SAVE REGS
003036 017700 000022 MOV @DLCNT,RO ; DELAY COUNT TO RO.
003042 001406 BEQ DLYCN ; BR IF 0.
003044 012701 000303 DLYAN: MOV #303,R1 ; 1 MSEC COUNT TO R1.
003050 005301 DLYBN: DEC R1 ; DECREMENT 1 MSEC COUNT.
003052 001376 BNE DLYBN ; BR IF NOT 0.
003054 005300 DEC RO ; DECREMENT DELAY COUNT.
003056 001372 BNE DLYAN ; BR IF NOT DONE DELAYING.
003060 104003 DLYCN: RSTO3
003062 000002 RTI ; EXIT.
003064 000000 DLCNT: OPEN
003066 000000 DLCTR: OPEN

; SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
003070 012737 177777 003136 INBINN: MOV #-1,RIND ; SET ALL VARIABLES
    
```

```

003076 013737 003136 003140      MOV      RIND,PIND
003104 013737 003136 003142      MOV      RIND,PTO
003112 013737 003136 003144      MOV      RIND,PT1
003120 013737 003136 003146      MOV      RIND,RT0
003126 013737 003136 003150      MOV      RIND,RT1
003134 000002                      RTI
003136 000000                      ;EXIT.
003140 000000      RIND: OPEN
003142 000000      PIND: OPEN
003144 000000      PTO: OPEN
003146 000000      PT1: OPEN
003150 000000      RT0: OPEN
003152 000000      RT1: OPEN
003154 000000      BINR: OPEN
003154 000000      BINP: OPEN
;SPECIAL BINARY COUNT PATTERN SUBROUTINE (PUNCH)
003156 013737 003142 003144      GTBINP: MOV      PTO,PT1      ;PREVIOUS BIN CHAR TO PT1
003164 005137 003144                      COM      PT1
003170 005137 003140                      COM      PIND
003174 001002                      BNE      .+6
003176 005237 003144                      INC      PT1
003202 043737 001346 003144      BIC      PCHMSK,PT1
003210 013737 003144 003142      MOV      PT1,PTO      ;SAVE BIN CHAR IN PTO
003216 013737 003144 003154      MOV      PT1,BINP     ;BIN CHAR TO BINP.
003224 013737 003144 001332      MOV      PT1,PCHOUT
003232 000002                      RTI
;SPECIAL BINARY COUNT PATTERN SUBROUTINE (READER)
003234 013737 003146 003150      GTBINR: MOV      RTO,RT1     ;PREVIOUS BIN CHAR TO RT1.
003242 005137 003150                      COM      RT1
003246 005137 003136                      COM      RIND
003252 001002                      BNE      .+6
003254 005237 003150                      INC      RT1
003260 043737 001350 003150      BIC      RDRMSK,RT1
003266 013737 003150 003146      MOV      RT1,RTO     ;SAVE BIN CHAR IN RTO.
003274 013737 003150 003152      MOV      RT1,BINR    ;BIN CHAR TO BINR.
003302 000002                      RTI
;EMT SUB TO CONVERT OCTAL TO ASCII.
003304 104006      OACNVV: SAVO55      ;SAVE REGS.
003306 013500                      MOV      2(5)+,R0     ;GET OCTAL VALUE.
003310 012501                      MOV      (5)+,R1      ;GET DESTINATION ADDR.
003312 012502                      MOV      (5)+,R2      ;GET CONVERT COUNT.
003314 060201                      ADD      R2,R1        ;DEVELOP ADDR TO STORE 1ST CHAR.
003316 010003      OACNVA: MOV      R0,R3
003320 042703 177770      BIC      #177770,R3   ;ISOLATE LEAST SIGNIFICANT DIGIT.
003324 062703 000060      ADD      #60,R3      ;CONVERT DIGIT TO ASCII.
003330 110341                      MOV      R3,-(1)     ;STORE ASCII CHARACTER.
003332 042700 000007      BIC      #7,R0
003336 006000                      ROR      R0
003340 006000                      ROR      R0
003342 006000                      ROR      R0
003344 005302                      DEC      R2
003346 001363      BNE      OACNVA      ;DONE ALL DIGITS?
003350 104007                      RSTO55      ;BRANCH IF NOT DONE.
003352 000002                      RTI          ;RESTORE REGS.
;EMT SUB TO MOVE VARIABLE NUMBER OF BYTES.
003354 104006      BMOVV: SAVO55      ;SAVE REGS.
003356 012501                      MOV      (5)+,R1     ;GET"FROM"ADDRESS
    
```

```

003360 012502          MOV      (5)+,R2          ;GET"TO"ADDRESS
003362 012503          MOV      (5)+,R3          ;GET COUNT
003364 112122          BMOVA:  MOVB     (1)+,(2)+      ;MOVE BYTE
003366 005303          DEC      R3              ;DECREMENT COUNT
003370 001375          BNE     BMOVA           ;BRANCH IF NOT DONE.
003372 104007          RSTOSS          ;RESTORE REGS.
003374 000002          RTI              ;DONE. EXIT.

;EMT SUB TO CONVERT BINARY TO DECIMAL ASCII.
BDCNVV: SAVOSS          ;SAVE REGS.
MOV      #DECVAL,RO     ;SET UP ADDR TO STORE DECIMAL ASCII IN RO
MOV      @ (5)+,R1      ;BINARY VALUE TO R1.
MOV      (5)+,BDCNVC    ;DESTINATION ADDR TO BDCNVC.
MOV      (5)+,BDCNVD    ;CHAR COUNT TO BDCNVD.
MOV      #ADTEMP,R2    ;ADDR OF TEN POWER STRING TO R2.
MOV      #5,CNVCTR      ;SET UP FOR 5 POWER CONVERSIONS.
BDCNVA: MOV      (2)+,TENPWR ;MOVE POWER OF TEN VALUE TO TENPWR.
JSR     R7,SUBTEN      ;PERFORM CONVERSION
DEC     CNVCTR          ;DONE 5 CONVERSIONS?
BNE     BDCNVA         ;BRANCH IF NOT YET 5.
SUB     BDCNVD,RO
MOV     RO,BDCNVB
BMOVE
BDCNVB: OPEN
BDCNVC: OPEN
BDCNVD: OPEN
RSTOSS          ;RESTORE REGS.
RTI              ;YES. EXIT.
SUBTEN: CLR     DIGIT    ;CLEAR DIGIT
SUBTNA: SUB     TENPWR,R1 ;SUBTRACT TEN POWER FROM BINARY VALUE.
BCS     SUBTNB        ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
INC     DIGIT
BR     SUBTNA
SUBTNB: ADD     TENPWR,R1 ;RESTORE SUBTRACTED VALUE.
ADD     #60,DIGIT      ;CONVERT (DIGIT) TO ASCII
MOVB   DIGIT,(0)+     ;MOVE ASCII CHAR TO DECVAL FIELD.
RTS     R7            ;EXIT.

CNVCTR: OPEN
DIGIT:  OPEN
TENPWR: OPEN
ADTEMP: 10000.
        1000.
        100.
        10.
        1.
003552 040 040 040 040 040 040 DECVAL: .BYTE 040,040,040,040,040,040
003555 040 040 040

;EMT RANDOM NUMBER GENERATOR. NUMBER IS STORED AT LOC AFTER SUB CALL.
RNGEN: SAVOSS
MOV     RP1,RO
ROL    RO
ROL    RO
ADD    RP2,RO
MOV    RO,RP1
ROL    RO
ROL    RO
ADD    RP2,RO
    
```



```

004050 005337 004150          DEC      BRCTR
004054 001375                BNE      .-4
004056 005077 175116          CLR      @PRB
                                ;CLEAR PTRI ENABLE.
004062 104016                ERRORN   ;TYPE NO PTR RESPONSE
004064 015337                EM7      ;MESSAGE
004066 177777                -1
004070 000755                BR      BREAD
                                ;TRY AGAIN.
004072 005077 175102  BREADA: CLR      @PRB
                                ;CLEAR READER CSR.
004076 017737 175100 001334  MOV      @PRB,CRBUF
                                ;CHAR READ TO CRBUF.
004104 000207                RTS      %7
                                ;EXIT SUBROUTINE.
004106 022626                POPSP2
                                ;RESTORE STACK.
004110 012737 000340 177776  MOV      #PRTY7,PSW
                                ;SET UP PRTY7.
004116 005777 175056          TST      @PRB
                                ;TEST FOR ERROR.
004122 100003                BPL      BREADC
                                ;BRANCH IF NO ERROR.
004124 004737 003732          JSR      PC,TSM2
004130 000735                BR      BREAD
004132 105777 175042  BREADC: TSTB   @PRB
                                ;TEST FOR DONE BIT.
004136 100755                BMI      BREADA
                                ;BRANCH IF DONE BIT SET.
004140 104016                ERRORN   ;ERROR.FALSE READER INTERRUPT.
004142 015361                EM10
004144 177777                -1
004146 000726                BR      BREAD
004150 000000                BRCTR: OPEN
                                ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
                                ;DETERMINED BY CONTENTS OF LOC STLMSK.
004152 104021                STAL:  RNDNUM
                                ;GET RANDOM NUMBER.
004154 000000                STLA:  OPEN
                                ;NUMBER GOES HERE.
004156 043737 004202 004154  BIC      STLMSK,STLA
                                ;APPLY STALL MASK.
004164 001405                BEQ      STALB
                                ;BRANCH IF RESULT IS 0.
004166 013737 004154 004176  MOV      STLA,STALA
004174 104031                DELAY
004176 000000                STALA: OPEN
                                ;DELAY COUNT
004200 000002                STALB: RTI
                                ;DONE. EXIT.
004202 000000                STLMSK: OPEN
                                ;STALL MASK.
                                ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT (1-77)
004204 104021                GRCNT: RNDNUM
                                ;GET RANDOM NUMBER.
004206 000000                GRCNTA: OPEN
                                ;NUMBER GOES HERE.
004210 043737 004230 004206  BIC      RCMSK,GRCNTA
                                ;APPLY MASK.
004216 001772                BEQ      GRCNT
                                ;TRY AGAIN IF RESULT 0
004220 013737 004206 004232  MOV      GRCNTA,RNCNT
                                ;COUNT TO RNCNT
004226 000207                RTS      %7
                                ;EXIT.
004230 000000                RCMSK: OPEN
                                ;RANDOM CHARACTER MASK.
004232 000000                RNCNT: OPEN
                                ;RANDOM CHARACTER COUNT.
                                ;SUB TO COMPARE DATA READ FROM READER AGAINST EXPECTED DATA AND REPORT ERRORS.
004234 104027                BCHECK: GETBNR
                                ;GET BIN CHARACTER
004236 023737 003152 001334  CMP      BINR,CRBUF
                                ;COMPARE BINR TO DATA IN CRBUF
004244 001001                BNE      .+4
                                ;BRANCH IF NOT SAME(ERROR).
004246 000207                RTS      %7
                                ;OK.EXIT.
004250 104017                OACNV
004252 003152                BINR
004254 015066                ASB
004256 000004                4
004260 104017                OACNV
004262 001334                CRBUF
004264 015101                AWAS
004266 000004                4

```

004270	104016		ERRORN		
004272	015043		EM1		
004274	177777		-1		
004276	005337	001344	DEC	ERCTR	; DECREMENT ERROR COUNTER
004302	001002		BNE	.+6	; BRANCH IF NO THIRD ERROR
004304	004737	004312	JSR	%7,BSYNC	; RESYNC THE READER.
004310	000207		RTS	%7	; EXIT.
			; SUBROUTINE TO SYNC THE READER TO A SPECIAL BINARY COUNT PATTERN TEST TAPE.		
004312	104026		BSYNC:	INBIN	; INITIALIZE BINARY PATTERN
004314	004737	004024	JSR	%7,BREAD	; READ CHAR.
004320	004737	004024	JSR	%7,BREAD	; READ CHAR.
004324	004737	004024	JSR	%7,BREAD	; READ CHAR AND STORE AT CHR1
004330	013737	001334	MOV	CRBUF,CHR1	
004336	004737	004024	JSR	%7,BREAD	; READ CHAR AND STORE AT CHR2
004342	013737	001334	MOV	CRBUF,CHR2	
004350	004737	004024	JSR	%7,BREAD	; READ CHAR AND STORE AT CHR3.
004354	013737	001334	MOV	CRBUF,CHR3	
004362	004737	004400	JSR	%7,SYNCA	; GO SYNC
004366	000751		BR	BSYNC	; NO SYNC. TRY AGAIN.
004370	012737	000003	MOV	#3,ERCTR	
004376	000207	001344	RTS	%7	; SUCCESS.EXIT.
004400	104004		SYNCA:	SAVOS	
004402	012700	001000	MOV	#512.,R0	; SET UP FOR 512 TRIES.
004406	104027		SYNCB:	GETBNR	; GET BIN CHAR.
004410	013701	003152	MOV	BINR,R1	; STORE AT R1.
004414	104027		GETBNR		; GET BIN CHAR.
004416	013702	003152	MOV	BINR,R2	; STORE AT R2.
004422	104027		GETBNR		; GET BIN CHAR.
004424	013703	003152	MOV	BINR,R3	; STORE AT R3.
004430	020137	001336	CMP	R1,CHR1	; MATCH ON 1ST WORD?
004434	001012		BNE	SYNCC	; BR IF NOT.
004436	020237	001340	CMP	R2,CHR2	; MATCH ON 2ND WORD?
004442	001007		BNE	SYNCC	; BR IF NOT.
004444	020337	001342	CMP	R3,CHR3	; MATCH ON 3RD WORD?
004450	001004		BNE	SYNCC	; BR IF NOT.
004452	104005		RSTOS		
004454	062716	000002	ADD	#2,(6)	; SET UP SINCE EXIT.
004460	000207		RTS	PC	; EXIT.
004462	005300		SYNCC:	DEC	R0
004464	001350		BNE	SYNCB	; TRIED 512 TIMES?
004466	104016		ERRORN		; BR IF NOT.
004470	015160		EM3		; YES. SYNC ERROR.
004472	177777		-1		
004474	104005		RSTOS		
004476	000207		RTS	PC	; SYNC ERROR EXIT.
			; SUBROUTINE TO CHECK FOR PUNCH READY.		
004500	005777	174500	CPRDY:	TST	@PPS
004504	100404		BMI	CPRDYA	; TEST FOR ERROR BIT.
004506	105777	174472	TSTB	@PPS	; BRANCH IF ERROR BIT SET.
004512	100001		BPL	CPRDYA	; TEST FOR READY BIT.
004514	000207		RTS	%7	; BRANCH IF READY NOT SET.
004516	104011		CPRDYA:	TYPES	; OK. EXIT.
004520	014647		SM3		; TYPE NOT READY MESSAGE.
004522	013254		IM16		
004524	177777		-1		
004526	104024		CHALT		

```

004530 000763
004532 004737 004500
004536 043737 001346 001332
004544 013777 001332 174434
004552 105777 174426
004556 100375
004560 000207

;SUBROUTINE TO PUNCH CHARACTER IN LOC PCHOUT.
HSPCH: JSR %7,CPRDY ;GO CHECK FOR PUNCH READY.
        BIC PCHMSK,PCHOUT
        MOV PCHOUT,@PPB ;LOAD PUNCH BUFFER.
        TSTB @PPS ;WAIT FOR DONE.
        BPL -4
        RTS %7 ;DONE. EXIT.

;SUBROUTINE TO SELECT PUNCH TO BE TESTED/USED.
PCHSEL: TYPESPCHM ;TYPE SELECT PUNCH MESSAGE.
        SPCHM
        IM23
        -1
        CHALT ;WAIT FOR USER.
        CMPB SR,PCHLIM ;VALID PUNCH NUMBER?
        BLO PCHSLA ;BR IF YES.
        TYPE ;NO. TYPE MESSAGE, AND
        INVRP ;GO TRY AGAIN.
        BR PCHSEL ;DEVELOP PCH CSR ADDR.
        MOV SR,PPS
        ASL PPS
        ASL PPS
        ADD FSTPCH,PPS
        MOV PPS,PPB ;DEVELOP PCH BUFFER ADDR.
        ADD #2,PPB
        MOV RDRLIM,PCHVTR ;DEVELOP PCH VECTOR ADDR.
        INC PCHVTR ;IF RDRLIM WAS ODD INCR TO MAKE IT EVEN.
        ROR PCHVTR
        ASL PCHVTR
        ADD SR,PCHVTR
        ASL PCHVTR
        ASL PCHVTR
        ADD FSTVCT,PCHVTR
        MOV #177400,PCHMSK ;SET UP PUNCH MASK FOR 8 LEVEL.
        TST SR ;8 LEVEL PUNCH?
        BPL PCHSLB ;BR IF YES.
        MOV #177700,PCHMSK ;NO. SET UP 6 LEVEL MASK.
        OACNV PCHSLB: ;CONVERT SELECTED PCH NUMBER TO ASCII.
        SR
        APCHID
        2
        TYPE ;TYPE PCH SELECTED MESSAGE.
        PCHIDM
        RTS PC ;EXIT.

;SUBROUTINE TO SELECT READER TO BE TESTED/USED.
RDRSEL: TYPESRDRM ;TYPE SELECT READER MESSAGE.
        SRDRM
        IM23
        -1
        CHALT ;WAIT FOR USER.
        CMPB SR,RDRLIM ;VALID READER?
        BLO RDRSLA ;BR IF YES.
        TYPE ;NO. TYPE MESSAGE AND TRY AGAIN.
        INVRP
        BR RDRSEL
        MOV SR,PRS ;DEVELOP RDR CSR ADDR.
  
```


005016	006337	001200		ASL	PRS	
005022	006337	001200		ASL	PRS	
005026	063737	001220	001200	ADD	FSTRDR, PRS	
005034	013737	001200	001202	MOV	PRS, PRB	; DEVELOP READER BUFFER ADDR.
005042	062737	000002	001202	ADD	#2, PRB	
005050	013737	177570	001210	MOV	SR, RDRVTR	; DEVELOP RDR VECTOR ADDR.
005056	006337	001210		ASL	RDRVTR	
005062	006337	001210		ASL	RDRVTR	
005066	063737	001244	001210	ADD	FSTVCT, RDRVTR	
005074	012737	177400	001350	MOV	#177400, RDRMSK	; SET UP 8 LEVEL READER MASK.
005102	005737	177570		TST	SR	; 8 LEVEL READER?
005106	100003			BPL	RDRSLB	; BR IF YES.
005110	012737	177700	001350	MOV	#177700, RDRMSK	; NO. SET UP 6 LEVEL MASK.
005116	104017			RDRSLB: OACNV		; CONVERT SELECTED RDR NUMBER TO ASCII.
005120	177570			SR		
005122	015606			ARDRID		
005124	000002			2		
005126	104010			TYPE		; TYPE RDR SELECTED MESSAGE.
005130	015601			RDRIDM		
005132	000207			RTS	PC	; EXIT.
005134	005037	001266		CLNUP: CLR	ICNT	; CLEAR ITERATION COUNT.
005140	005037	001262		CLR	RTNNO	; CLEAR CURRENT ROUTINE NUMBER.
005144	012737	000003	001344	MOV	#3, ERCTR	; SET ERROR COUNT TO 3.
005152	012701	000300		MOV	#300, R1	; CLEAR INTERRUPT VECTORS.
005156	012702	000302		MOV	#302, R2	
005162	010221			CLNUPA: MOV	R2, (1)+	
005164	005021			CLR	(1)+	
005166	020237	001000		CMP	R2, 1000	
005172	001403			BEQ	CLNUPB	
005174	062702	000004		ADD	#4, R2	
005200	000770			BR	CLNUPA	
005202	000207			CLNUPB: RTS	PC	; EXIT.

```

;PRGO - READER TESTS
X=-1
Y=0
005204 012737 005226 001256 PRGO: MOV #POTO,KSTART ;ADDR OF 1ST ROUTINE TO KSTART.
005212 104010 ;TYPE TITLE.
005214 013227 ;TITLO
005216 004737 004760 JSR PC,RDRSEL ;SELECT READER.
005222 000137 001640 JMP SRSET ;GO GET STARTED.
005226 TSTA POA,1000.
005226 TSTAA POA,1000.,\X+1,\X+2,\Y
;*****
POTO: 0 ; PRGO ROUTINE 0 *
005226 000000 ; ADDRESS OF NEXT ROUTINE *
005230 005256 ; TEST ITERATION COUNT *
005232 001750 ; SCOPE ENTRY POINT *
005234 005244
000000 X=X+1
;*****
;TEST ABILITY TO REFERENCE THE READER STATUS WORD
005236 012737 005252 000004 POAA: MOV #POAE,MACHER ;SET UP MACHINE ERROR TRAP.
005244 005777 173730 ;REFERENCE READER STATUS WORD.
005250 104000 ;SCOPE
005252 104015 POAE: ERROR ;ERROR. TRAPPED WHEN REFERENCING READER
005254 104000 ;SCOPE ;STATUS WORD (PRS).
005256 TSTA POB,1000.
005256 TSTAA POB,1000.,\X+1,\X+2,\Y
;*****
POT1: 1 ; PRGO ROUTINE 1 *
005256 000001 ; ADDRESS OF NEXT ROUTINE *
005260 005306 ; TEST ITERATION COUNT *
005262 001750 ; SCOPE ENTRY POINT *
005264 005274
000001 X=X+1
;*****
;TEST ABILITY TO REFERENCE THE READER BUFFER.
005266 012737 005302 000004 POBA: MOV #POBB,MACHER ;SET UP MACHINE ERROR TRAP.
005274 005777 173702 ;REFERENCE READER BUFFER
005300 104000 ;SCOPE
005302 104015 POBB: ERROR ;ERROR. TRAPPED WHEN REFERENCING
005304 104000 ;SCOPE ;READER BUFFER. (PRB)
005306 TSTA POD,1000.
005306 TSTAA POD,1000.,\X+1,\X+2,\Y
;*****
POT2: 2 ; PRGO ROUTINE 2 *
005306 000002 ; ADDRESS OF NEXT ROUTINE *
005310 005362 ; TEST ITERATION COUNT *
005312 001750 ; SCOPE ENTRY POINT *
005314 005316
000002 X=X+1
;*****
;TEST ABILITY TO SET AND CLEAR THE ID BIT (INTERRUPT ENABLE (BIT 6))
;IN READER STATUS WORD
005316 052777 000100 173654 PODA: BIS #BIT6,PRB ;SET ID BIT (BIT 6) IN READER PRS
005324 032777 000100 173646 BIT #BIT6,PRB ;CHECK ID BIT IN PRS
005332 001002 BNE POBB ;ID BIT SET?
005334 104015 ERROR ;NO. ERROR. FAILED TO SET ID BIT (BIT 6)
;IN PRS.
005336 104000 SCOPE
  
```

```

005340 042777 000100 173632 PODB: BIC #BIT6, @PRS ;CLEAR ID BIT IN PRS.
005346 032777 000100 173624 BIT #BIT6, @PRS ;CHECK ID BIT IN PRS
005354 001401 BEQ .+4 ;BRANCH IF BIT CLEAR.
005356 104015 ERROR ;ERROR. ID BIT IN PRS FAILED TO CLEAR.
005360 104000 SCOPE
005362 TSTA POE, 100.
005362 TSTAA POE, 100., \X+1, \X+2, \Y
;*****
005362 000003 POT3: 3 ; PRGO ROUTINE 3 *
005364 005416 POT4 ; ADDRESS OF NEXT ROUTINE *
005366 000144 100. ; TEST ITERATION COUNT *
005370 005372 POEA ; SCOPE ENTRY POINT *
000003 X=X+1
;*****
005372 052777 000100 173600 POEA: BIS #BIT6, @PRS ;SET ID BIT IN PRS
005400 104001 SRESET ;RESET
005402 032777 000100 173570 BIT #BIT6, @PRS ;TEST ID BIT
005410 001401 BEQ .+4 ;BRANCH IF ID BIT IS CLEAR.
005412 104015 ERROR ;ERROR. RESET INSTRUCTION FAILED TO
005414 104000 SCOPE ;CLEAR ID BIT IN READER PRS.
005416 TSTA POF, 100.
005416 TSTAA POF, 100., \X+1, \X+2, \Y
;*****
005416 000004 POT4: 4 ; PRGO ROUTINE 4 *
005420 005450 POTS ; ADDRESS OF NEXT ROUTINE *
005422 000144 100. ; TEST ITERATION COUNT *
005424 005426 POFA ; SCOPE ENTRY POINT *
000004 X=X+1
;*****
005426 005277 173546 POFA: INC @PRS ;ENABLE READER
005432 104031 DELAY ;WAIT APPROX 100 MILLISECS.
005434 000144 100.
005436 105777 173536 TSTB @PRS ;TEST FOR DONE (BIT 7)
005442 100401 BMI POFB ;BRANCH IF DONE BIT WAS SET.
005444 104015 ERROR ;ERROR. 100 MSECS AFTER READER
;ENABLE, DONE BIT WAS NOT SET.
005446 104000 POFB: SCOPE ;SCOPE
005450 TSTA POG, 1000.
005450 TSTAA POG, 1000., \X+1, \X+2, \Y
;*****
005450 000005 POTS: 5 ; PRGO ROUTINE 5 *
005452 005502 POT6 ; ADDRESS OF NEXT ROUTINE *
005454 001750 1000. ; TEST ITERATION COUNT *
005456 005460 POGA ; SCOPE ENTRY POINT *
000005 X=X+1
;*****
005460 005277 173514 POGA: INC @PRS ;ENABLE READER
005464 104031 DELAY ;WAIT APPROX 100 MILLISECS
005466 000144 100.
005470 105777 173504 TSTB @PRS ;TEST DONE BIT (BIT 7 OF PRS)
005474 100401 BMI .+4 ;BRANCH IF DONE BIT SET
005476 104015 ERROR ;ERROR. DONE BIT NOT SET, OR FAILED
005500 104000 SCOPE ;TO READ IT.

```

```

005502          TSTA      POH,100.
005502          TSTAA     POH,100.,\X+1,\X+2,\Y
;*****
005502 000006   POT6:    6          ; PRGO ROUTINE 6          *
005504 005552   POT7     POT7       ; ADDRESS OF NEXT ROUTINE *
005506 000144   100.      ; TEST ITERATION COUNT   *
005510 005512   POHA     POHA       ; SCOPE ENTRY POINT      *
          X=X+1
;*****
005512 005277 173462  POHA:    INC      @PRS      ; TEST THAT RESET COMMAND CLEARS DONE BIT (BIT 7 OF PRS)
005516 104031   DELAY     ; ENABLE READER
005520 000062   50.      ; DELAY APPROX 50 MILLISECONDS
005522 105777 173452   TSTB     @PRS      ; TEST FOR DONE BIT
005526 100005   BPL      POHB     ; BRANCH IF DONE BIT NOT SET
005530 000005   RESET    ; RESET
005532 105777 173442   TSTB     @PRS      ; TEST DONE BIT
005536 100403   BMI      POHC     ; BRANCH IF DONE BIT STILL SET.
005540 104000   SCOPE    ; SCOPE
005542 104015   POHB:    ERROR    ; ERROR 1. DONE BIT NOT SET.
005544 104000   SCOPE    ;
005546 104015   POHC:    ERROR    ; ERROR 2. DONE BIT NOT RESET BY
005550 104000   SCOPE    ; RESET INSTRUCTION.
005552          TSTA      POI,100.
005552          TSTAA     POI,100.,\X+1,\X+2,\Y
;*****
005552 000007   POT7:    7          ; PRGO ROUTINE 7          *
005554 005614   POT10    POT10     ; ADDRESS OF NEXT ROUTINE *
005556 000144   100.     ; TEST ITERATION COUNT   *
005560 005562   POIA     POIA     ; SCOPE ENTRY POINT      *
          X=X+1
;*****
005562 104001   POIA:    SRESET   ; TEST THAT DONE BIT (BIT 7 OF PRS) IS CLEARED WHEN ENABLING THE READER
005564 005277 173410   INC      @PRS      ; RESET
005570 105777 173404   TSTB     @PRS      ; ENABLE READER
005574 100375   BPL      -4       ; TEST FOR DONE BIT
005576 005277 173376   INC      @PRS      ; BRANCH IF DONE BIT NOT SET
005602 105777 173372   TSTB     @PRS      ; ENABLE READER AGAIN
005606 100001   BPL      +4       ; TEST DONE BIT AGAIN
005610 104015   ERROR    ; BRANCH IF DONE BIT IS RESET
005612 104000   SCOPE    ; READER ENABLE DID NOT CLEAR DONE BIT
005614          TSTA      POJ,100.
005614          TSTAA     POJ,100.,\X+1,\X+2,\Y
;*****
005614 000010   POT10:  10         ; PRGO ROUTINE 10        *
005616 005654   POT11    POT11     ; ADDRESS OF NEXT ROUTINE *
005620 000144   100.     ; TEST ITERATION COUNT   *
005622 005624   POJA     POJA     ; SCOPE ENTRY POINT      *
          X=X+1
;*****
005624 005277 173350   POJA:    INC      @PRS      ; TEST THAT DONE BIT IS CLEARED BY REFERENCING READER BUFFER (PRB)
005630 105777 173344   TSTB     @PRS      ; ENABLE READER
005634 100375   BPL      -4       ; TEST FOR DONE BIT
005636 005777 173340   TST      @PRB     ; BRANCH IF DONE BIT NOT SET.
          ; REFERENCE READER BUFFER (PRB)
  
```

```

005642 105777 173332      TSTB      @PRS      ;TEST FOR DONE BIT
005646 100001      BPL        .+4      ;BRANCH IF DONE BIT IS CLEAR.
005650 104015      ERROR     ;ERROR 1. DONE BIT WAS NOT CLEARED
005652 104000      SCOPE     ;BY REFERENCING READER BUFFER.
005654      TSTA      POM,100.
005654      TSTAA     POM,100.,\X+1,\X+2,\Y
;*****
POT11: 11      ; PRGO ROUTINE 11 *
005654 000011      POT12     ;ADDRESS OF NEXT ROUTINE *
005656 005756      100.      ;TEST ITERATION COUNT *
005660 000144      POMA      ;SCOPE ENTRY POINT *
005662 005664      X=X+1
;*****
;TEST ABILITY TO READ READER BUFFER RELIABLY.
POMA:  MOV      #100,%D      ;SET COUNT TO 100 IN RD
005664 012700 000144      JSR      %7,AREAD      ;GET CHARACTER
005670 004737 003654      MOV      @PRB,CHR1     ;C(PRB) TO CHR1
005674 017737 173302 001336  POMB:  MOV      @PRB,CHR2     ;C(PRB) TO CHR2
005702 017737 173274 001340      CMP      CHR1,CHR2     ;COMPARE CHR1 AND CHR2.
005710 023737 001336 001340      BNE      POMC          ;BRANCH IF R1 AND R2 DON'T MATCH
005716 001003
005720 005300
005722 001367
005724 104000      POMC:  SCOPE          ;SCOPE
005726 104017      OACNV     ;CORRECT 1ST READ DATA TO ASCII
005730 001336      CHR1
005732 015140      ORGRD
005734 000004      4
005736 104017      OACNV
005740 001340      CHR2
005742 015153      SUBRD
005744 000004      4
005746 104016      ERRORN     ;ERROR. REREAD OF PRB DID NOT MATCH
005750 015106      EM2       ;INITIAL DATA READ FROM PRB.
005752 177777      -1
005754 104000      SCOPE
005756      TSTA      PON,100.
005756      TSTAA     PON,100.,\X+1,\X+2,\Y
;*****
POT12: 12      ; PRGO ROUTINE 12 *
005760 006020      POT13     ;ADDRESS OF NEXT ROUTINE *
005762 000144      100.      ;TEST ITERATION COUNT *
005764 005766      PONA      ;SCOPE ENTRY POINT *
005766 000012      X=X+1
;*****
;TEST THAT READER BUFFER (PRB) IS CLEARED BY READER ENABLE
PONA:  SRESET      ;RESET
005770 004737 003654      JSR      %7,AREAD     ;GET CHARACTER
005774 005777 173202      TST      @PRB        ;TEST CONTENTS OF READER BUFFER.
006000 001772      BEQ      PONA        ;GO GET ANOTHER CHAR IF 0.
006002 005277 173172      INC      @PRS        ;NOT 0. ENABLE READER
006006 005777 173170      TST      @PRB        ;CHECK PRB
006012 001401      BEQ      .+4         ;BRANCH IF PRB IS RESET
006014 104015      ERROR     ;ERROR. PRB NOT RESET BY READER ENABLE.
006016 104000      SCOPE
006020      TSTA      P00,100.
006020      TSTAA     P00,100.,\X+1,\X+2,\Y

```

```

006020 000013          ;*****
006022 006070          POT13: 13          ; PRGO ROUTINE 13          *
006024 000144          ; POT14          ; ADDRESS OF NEXT ROUTINE  *
006026 006034          ; 100.          ; TEST ITERATION COUNT    *
          000013          ; POQA          ; SCOPE ENTRY POINT      *
          X=X+1          ;*****
          ;TEST THAT READER IS ABLE TO INTERRUPT. IF INTERRUPT IS SERVICED, IT WILL
          ;HAVE OCCURRED AT CORRECT VECTOR.
006030 104013          STRDRV          ;SET UP READER INTERRUPT VECTOR
006032 006066          POQB
006034 012737 000000 177776 POQA: MOV #PRTYO,PSW ;SET PROCESSOR PRIORITY TO 0
006042 042777 000100 173130 BIC #BIT6,@PRS ;DISABLE READER INTERRUPT.
006050 004737 003654 JSR %7,AREAD ;GO READ CHARACTER.
006054 052777 000100 173116 BIS #BIT6,@PRS ;ENABLE READER INTERRUPT.
006062 000240 NOP ;NO OP
006064 104015 ERROR ;ERROR. READER FAILED TO INTERRUPT.
006066 104000 POQB: SCOPE ;SCOPE
006070 TSTA POP,100.
006070 TSTA POP,100.,\X+1,\X+2,\Y
          ;*****
006070 000014          POT14: 14          ; PRGO ROUTINE 14          *
006072 006144          ; POT15          ; ADDRESS OF NEXT ROUTINE  *
006074 000144          ; 100.          ; TEST ITERATION COUNT    *
006076 006104          ; POPA          ; SCOPE ENTRY POINT      *
          000014          ; X=X+1          ;*****
          ;TEST THAT READER DOES NOT INTERRUPT WITH PROCESSOR AT SAME PRIORITY
          ;LEVEL AS READER.
006100 104013          STRDRV          ;SET UP READER INTERRUPT VECTOR
006102 006140          POPE
006104 013737 001212 177776 POPA: MOV RDRVLV,PSW ;SET PROCESSOR PRIORITY SAME AS READER PRIORITY.
006112 005077 173062 CLR @PRS ;DISABLE READER INTERRUPT.
006116 004737 003654 JSR %7,AREAD ;GO READ A CHARACTER.
006122 052777 000100 173050 BIS #BIT6,@PRS ;ENABLE READER INTERRUPT.
006130 000240 NOP ;OK IF NO INTERRUPT OCCURS.
006132 005077 173042 CLR @PRS ;DISABLE READER INTERRUPT.
006136 104000 SCOPE ;SCOPE
006140 104015 POPE: ERROR ;ERROR. READER ERRONEOUSLY INTERRUPTED
          ;WITH PROCESSOR AT SAME PRIORITY LEVEL AS
          ;THE READER, OR THE READER IS AT HIGHER
          ;PRIORITY LEVEL THAN SPECIFIED AT RDRVLV.
006142 104000          SCOPE
006144 TSTA POP,100.
006144 TSTA POP,100.,\X+1,\X+2,\Y
          ;*****
006144 000015          POT15: 15          ; PRGO ROUTINE 15          *
006146 006222          ; POT16          ; ADDRESS OF NEXT ROUTINE  *
006150 000144          ; 100.          ; TEST ITERATION COUNT    *
006152 006160          ; POQA          ; SCOPE ENTRY POINT      *
          000015          ; X=X+1          ;*****
          ;TEST THAT READER INTERRUPTS WITH PROCESSOR AT PRIORITY 1 LEVEL LOWER
          ;THAN READER'S
006154 104013          STRDRV          ;SET UP READER INTERRUPT VECTOR
006156 006220          POQB
006160 013737 001212 177776 POQA: MOV RDRVLV,PSW ;SET PROCESSOR PRIORITY ONE LEVEL LOWER

```

```

006166 162737 000040 177776 SUB #40,PSW ;THAN READER PRIORITY
006174 042777 000100 172776 BIC #BIT6,@PRS ;DISABLE READER INTERRUPT
006202 004737 003654 JSR %7,AREAD ;GO READ A CHARACTER.
006206 052777 000100 172764 BIS #BIT6,@PRS ;ENABLE READER INTERRUPT
006214 000240 NOP ;NOP
006216 104015 ERROR ;READER FAILED TO INTERRUPT WITH
;PROCESSOR PRIORITY ONE LEVEL LOWER THAN
;READER. THEREFORE, READER PRIORITY MUST BE
;LOWER THAN SPECIFIED AT RDRLVL

006220 104000 PQQB: SCOPE
006222 TSTA POR,100.
006222 TSTAA POR,100.,\X+1,\X+2,\Y
;*****
POT16: 16 ; PRGD ROUTINE 16 *
; POT17 ; ADDRESS OF NEXT ROUTINE *
; 100. ; TEST ITERATION COUNT *
; PORA ; SCOPE ENTRY POINT *
; X=X+1
;*****
;TEST THAT READER DOES NOT REINTERRUPT AFTER RTI WHEN DONE BIT IS NOT CLEARED
PORA: STRDRV ;SET READER INTERRUPT VECTOR
PORC
MOV #PRTY0,PSW ;SET PROCESSOR TO PRIORITY 0
CLR @PRS ;DISABLE READER INTERRUPT.
JSR %7,AREAD ;GO READ A CHARACTER.
BIS #BIT6,@PRS ;ENABLE READER INTERRUPT
NOP
ERROR ;ERROR 1. READER FAILED TO INTERRUPT
SCOPE
PORC: MOV #PORE,@RDVTR ;CHANGE INTERRUPT VECTOR TO PORE
MOV #PORD,@%6
RTI ;RETURN FROM INTERRUPT
PORD: NOP
SCOPE
PORE: ERROR ;ERROR 2. READER REINTERRUPTED AFTER
SCOPE ;RTI WITH DONE BIT LEFT ON
TSTA POS,1000.
TSTAA POS,1000.,\X+1,\X+2,\Y
;*****
POT17: 17 ; PRGD ROUTINE 17 *
; POT20 ; ADDRESS OF NEXT ROUTINE *
; 1000. ; TEST ITERATION COUNT *
; POSA ; SCOPE ENTRY POINT *
; X=X+1
;*****
;TEST THAT READER INTERRUPTS IMMEDIATELY UPON LOWERING CP PRIORITY TO 0.
;SET READER INTERRUPT VECTOR TO
POSA: STRDRV
POSB
CLR @PRS ;DISABLE PTRI.
JSR %7,AREAD ;READ A CHARACTER.
BIS #BIT6,@PRS ;ENABLE PTRI
CLR PSW ;LOWER PRIORITY TO 0.
MOV #PRTY7,PSW ;RAISE PRIORITY BACK TO 7.
ERROR ;ERROR. READER FAILED TO INTERRUPT IMMEDIATELY
;AFTER LOWERING PRIORITY TO 0
;INTERRUPTS TO HERE IF SUCCESSFUL.

006324 104013 POSB: SCOPE
006326 006362 TSTA POT,10000.
006330 005077 172644 POSA: CLR @PRS ;DISABLE PTRI.
006334 004737 003654 JSR %7,AREAD ;READ A CHARACTER.
006340 052777 000100 172632 BIS #BIT6,@PRS ;ENABLE PTRI
006346 005037 177776 CLR PSW ;LOWER PRIORITY TO 0.
006352 012737 000340 177776 MOV #PRTY7,PSW ;RAISE PRIORITY BACK TO 7.
006360 104015 ERROR ;ERROR. READER FAILED TO INTERRUPT IMMEDIATELY
;AFTER LOWERING PRIORITY TO 0
;INTERRUPTS TO HERE IF SUCCESSFUL.

006362 104000 POSB: SCOPE
006364 TSTA POT,10000.

```

006364

006364 000020
006366 006412
006370 023420
006372 006400
000020

```
TSTAA POT,10000.,\X+1,\X+2,\Y
:*****
POT20: 20 ; PRGO ROUTINE 20 *
        POT21 ; ADDRESS OF NEXT ROUTINE *
        10000. ; TEST ITERATION COUNT *
        POTA ; SCOPE ENTRY POINT *
        X=X+1
```

006374 004737 004312
006400 004737 004024
006404 004737 004234
006410 104000
006412
006412

```
:*****
;READ AND CHECK 10000 CHARACTERS OF SPECIAL BINARY COUNT PATTERN. FULL SPEED.
POTA: JSR %7,BSYNC ; SYNC READER; SET ERROR COUNTER.
        JSR %7,BREAD ; GO READ CHARACTER
        JSR %7,BCHECK ; GO CHECK CHARACTER READ.
        SCOPE ; SCOPE
        TSTA POU,500.
        TSTAA POU,500.,\X+1,\X+2,\Y
```

006412 000021
006414 006450
006416 000764
006420 006434
000021

```
:*****
POT21: 21 ; PRGO ROUTINE 21 *
        POT22 ; ADDRESS OF NEXT ROUTINE *
        500. ; TEST ITERATION COUNT *
        POUA ; SCOPE ENTRY POINT *
        X=X+1
```

006422 012737 177770 004202
006430 004737 004312
006434 104012
006436 004737 004024
006442 004737 004234
006446 104000
006450
006450

```
:*****
;READ AND CHECK 500 CHARACTERS OF SPECIAL BINARY COUNT PATTERN.
;RANDOM STALL BETWEEN CHARACTERS (0 TO 7 MSECS).
POUA: MOV #177770,STLMSK ; LIMIT STALLS TO 31 MSECS.
        JSR %7,BSYNC ; SYNC READER; SET ERROR COUNTER
        STALL ; RANDOM STALL (0 TO 7 MSECS)
        JSR %7,BREAD ; GO READ CHARACTER
        JSR %7,BCHECK ; GO CHECK CHARACTER READ
        SCOPE ; SCOPE
        TSTA POV,1000.
        TSTAA POV,1000.,\X+1,\X+2,\Y
```

006450 000022
006452 006522
006454 001750
006456 006472
000022

```
:*****
POT22: 22 ; PRGO ROUTINE 22 *
        POT23 ; ADDRESS OF NEXT ROUTINE *
        1000. ; TEST ITERATION COUNT *
        POVA ; SCOPE ENTRY POINT *
        X=X+1
```

006460 012737 177740 004202
006466 004737 004312
006472 012737 000003 004232
006500 104012
006502 004737 004024
006506 004737 004234
006512 005337 004232
006516 001371
006520 104000
006522
006522

```
:*****
;READ 1000 GROUPS OF 3 CHARS EACH. STALL (0 TO 31 MSECS) BEFORE EACH GROUP.
POVA: MOV #177740,STLMSK ; LIMIT STALLS TO 31 MSECS.
        JSR %7,BSYNC ; SYNC READER. SET ERROR COUNTER
        MOV #3,RNCNT ; SET CHAR COUNT TO 3.
        STALL ; RANDOM STALL (0 TO 31 MSECS).
POVB: JSR %7,BREAD ; GO READ CHARACTER.
        JSR %7,BCHECK ; GO CHECK CHARACTER READ.
        DEC RNCNT ; 3 CHARS READ?
        BNE POVB ; BR IF NOT 3 CHARS YET.
        SCOPE ; SCOPE
        TSTA POX,1000.
        TSTAA POX,1000.,\X+1,\X+2,\Y
```

006522 000023
006524 006600
006526 001750
006530 006552

```
:*****
POT23: 23 ; PRGO ROUTINE 23 *
        POT24 ; ADDRESS OF NEXT ROUTINE *
        1000. ; TEST ITERATION COUNT *
        POXA ; SCOPE ENTRY POINT *
```


000023

```

X=X+1
;*****
;READ AND CHECK 1000 CHARACTER GROUPS OF RANDOM LENGTH (1 TO 15).
;RANDOM STALL (0 TO 31 MSECS) BETWEEN GROUPS.
MOV #177740,STLMSK ;LIMIT STALLS TO 31 MSECS.
MOV #177760,RCMSK ;LIMIT MAY CHAR COUNT TO 15 CHARS.
JSR %7,BSYNC ;SYNC READER. SET ERROR COUNTER.
POXA: JSR %7,GRCNT ;GENERATE RANDOM CHAR COUNT.
STALL
POXB: JSR %7,BREAD ;GO READ CHARACTER.
JSR %7,BCHECK ;GO CHECK CHARACTER.
DEC RNCNT ;ALL CHARS READ?
BNE POXB ;BRANCH IF NOT.
SCOPE
TSTAA POY,1000.,\X+1,LST,\Y

```

```

006532 012737 177740 004202
006540 012737 177760 004230
006546 004737 004312
006552 004737 004204
006556 104012
006560 004737 004024
006564 004737 004234
006570 005337 004232
006574 001371
006576 104000
006600

```

```

;*****
POT24: 24 ; PRG0 ROUTINE 24 *
POTLST ; ADDRESS OF NEXT ROUTINE *
1000. ; TEST ITERATION COUNT *
POYA ; SCOPE ENTRY POINT *
X=X+1

```

```

006600 000024
006602 177777
006604 001750
006606 006630
000024

```

```

;*****
;READ AND CHECK 1000 CHARACTER GROUPS OF SPECIAL BINARY COUNT PATTERN.
;RANDOM LENGTH
;GROUPS (BETWEEN 1 AND 77). RANDOM STALL BETWEEN GROUPS (0 TO 31 MSECS).

```

```

006610 012737 177740 004202
006616 012737 177700 004230
006624 004737 004312
006630 004737 004204
006634 104012
006636 004737 004024
006642 004737 004234
006646 005337 004232
006652 001371
006654 104000

```

```

MOV #177740,STLMSK
MOV #177700,RCMSK
JSR %7,BSYNC ;SYNC READER; SET ERROR COUNTER.
POYA: JSR %7,GRCNT ;GENERATE RANDOM CHARACTER COUNT.
STALL ;RANDOM STALL (0 TO 31MSECS)
POYB: JSR %7,BREAD ;GO READ CHARACTER
JSR %7,BCHECK ;GO CHECK CHARACTER READ
DEC RNCNT ;DECREMENT RANDOM CHAR COUNT
BNE POYB ;GO READ AGAIN IF COUNT NOT 0.
SCOPE
;PRG1 - PUNCH TESTS
X=-1
Y=1

```

```

177777
000001
006656 012737 006700 001256
006664 104010
006666 013111
006670 004737 004562
006674 000137 001640
006700
006700

```

```

;ADDR OF 1ST ROUTINE TO KSTART
;TYPE TITLE.
JSR PC,PCHSEL ;SELECT PUNCH.
JMP SRSET ;GO GET STARTED.
TSTA PIA,1000.
TSTAA PIA,1000.,\X+1,\X+2,\Y

```

```

006700 000000
006702 006730
006704 001750
006706 006716
000000

```

```

;*****
PITO: 0 ; PRG1 ROUTINE 0 *
PITI ; ADDRESS OF NEXT ROUTINE *
1000. ; TEST ITERATION COUNT *
PIAA ; SCOPE ENTRY POINT *
X=X+1

```

```

006710 012737 006724 000004
006716 005777 172262
006722 104000

```

```

;*****
;TEST ABILITY TO REFERENCE THE PUNCH STATUS WORD (PPS)
PIAA: MOV #PIAB,MACHER
TST @PPS ;REFERENCE PUNCH STATUS WORD
SCOPE ;SCOPE

```

006724 104015
006726 104000
006730
006730

P1AB: ERROR ;ERROR. TRAPPED WHEN REFERENCING PUNCH
SCOPE ;STATUS WORD (PPS).
TSTA P1B,1000.
TSTAA P1B,1000.,\X+1,\X+2,\Y

006730 000001
006732 006760
006734 001750
006736 006746
000001

P1T1: 1 ; PRG1 ROUTINE 1 *
PIT2 ; ADDRESS OF NEXT ROUTINE *
1000. ; TEST ITERATION COUNT *
P1BA ; SCOPE ENTRY POINT *
X=X+1

006740 012737 006754 000004
006746 005777 172234
006752 104000
006754 104015
006756 104000
006760
006760

;TEST ABILITY TO REFERENCE THE PUNCH BUFFER (PPB)
MOV #P1BB,MACHER ;SET UP MACHINE ERROR TRAP.
P1BA: TST @PPB ;REFERENCE PUNCH BUFFER.
SCOPE ;SCOPE
P1BB: ERROR ;TRAPPED WHEN REFERENCING
SCOPE ;PUNCH BUFFER (PPB)
TSTA PIC,1000.
TSTAA PIC,1000.,\X+1,\X+2,\Y

006760 000002
006762 007034
006764 001750
006766 006770
000002

P1T2: 2 ; PRG1 ROUTINE 2 *
PIT3 ; ADDRESS OF NEXT ROUTINE *
1000. ; TEST ITERATION COUNT *
P1CA ; SCOPE ENTRY POINT *
X=X+1

006770 052777 000100 172206
006776 032777 000100 172200
007004 001002
007006 104015
007010 104000
007012 042777 000100 172164
007020 032777 000100 172156
007026 001401
007030 104015
007032 104000
007034

;TEST ABILITY TO SET AND CLEAR ID BIT (BIT 6) IN PPS
P1CA: BIS #BIT6,@PPS ;SET ID BIT IN PPS (BIT 6)
BIT #BIT6,@PPS ;CHECK ID BIT IN PPS
BNE PICB ;BRANCH IF BIT SET
ERROR. FAILED TO SET ID BIT (BIT 6) IN
SCOPE PPS
P1CB: BIC #BIT6,@PPS ;CLEAR ID BIT IN PPS
BIT #BIT6,@PPS ;CHECK ID BIT IN PPS
BEQ .+4 ;BRANCH IF BIT IS CLEAR
ERROR. ID BIT IN PPS FAILED TO CLEAR
SCOPE
TSTA PID,100.
TSTAA PID,100.,\X+1,\X+2,\Y

007034 000003
007036 007070
007040 000144
007042 007044
000003

P1T3: 3 ; PRG1 ROUTINE 3 *
PIT4 ; ADDRESS OF NEXT ROUTINE *
100. ; TEST ITERATION COUNT *
P1DA ; SCOPE ENTRY POINT *
X=X+1

007044 052777 000100 172132
007052 104001
007054 032777 000100 172122
007062 001401
007064 104015
007066 104000
007070
007070

;TEST ABILITY TO CLEAR ID BIT (6) IN PPS
P1DA: BIS #BIT6,@PPS ;SET ID BIT IN PPS.
SRESET ;RESET.
BIT #BIT6,@PPS ;TEST ID BIT IN PPS.
BEQ .+4 ;BRANCH IF ID BIT IS CLEAR.
ERROR. RESET INSTRUCTION FAILED TO
SCOPE ;CLEAR ID BIT (BIT 6) IN PPS.
TSTA PIE,1000.
TSTAA PIE,1000.,\X+1,\X+2,\Y

```

007070 000004          PIT4: 4          ; PRG1 ROUTINE 4          *
007072 007112          PIT5          ; ADDRESS OF NEXT ROUTINE *
007074 001750          1000.          ; TEST ITERATION COUNT   *
007076 007100          PIEA          ; SCOPE ENTRY POINT      *
          000004          X=X+1

;*****
;TEST THAT READY BIT (BIT 7) IS SET FOLLOWING A RESET INSTRUCTION, AND
;THAT THE READY BIT CAN BE READ RELIABLY.
007100 105777 172100  PIEA: TSTB  @PPS          ; TEST PPS
007104 100401          BMI          .+4          ; BRANCH IF READY BIT SET
007106 104015          ERROR          ; ERROR. RESET FAILED TO SET READY BIT,
007110 104000          SCOPE          ; OR FAILED TO READ READY BIT.
007112          TSTA  PIF,100.
007112          TSTAA PIF,100.,\X+1,\X+2,\Y

;*****
007112 000005          PIT5: 5          ; PRG1 ROUTINE 5          *
007114 007150          PIT6          ; ADDRESS OF NEXT ROUTINE *
007116 000144          100.          ; TEST ITERATION COUNT   *
007120 007122          PIFA          ; SCOPE ENTRY POINT      *
          000005          X=X+1

;*****
;TEST THAT READY BIT (BIT 7) OF PPS IS RESET BY LOADING PUNCH BUFFER (PPB)
007122 104001          PIFA: SRESET          ; RESET
007124 004737 004500 JSR  %7,CPRDY          ; CHECK FOR PUNCH READY
007130 112777 000000 172050 MOVB #0,@PPB          ; LOAD 0 INTO PUNCH BUFFER (PPB)
007136 105777 172042 TSTB @PPS          ; TEST PPS
007142 100001          BPL          .+4          ; BRANCH IF READY BIT CLEAR
007144 104015          ERROR          ; ERROR. LOADING PUNCH BUFFER (PPB)
007146 104000          SCOPE          ; FAILED TO RESET READY BIT IN PPS
007150          TSTA  PIG,100.
007150          TSTAA PIG,100.,\X+1,\X+2,\Y

;*****
007150 000006          PIT6: 6          ; PRG1 ROUTINE 6          *
007152 007212          PIT7          ; ADDRESS OF NEXT ROUTINE *
007154 000144          100.          ; TEST ITERATION COUNT   *
007156 007160          PIGA          ; SCOPE ENTRY POINT      *
          000006          X=X+1

;*****
;TEST THAT READY BIT (BIT 7) IS NOT RESET BY BYTE LOADING PPB+1.
007160 104001          PIGA: SRESET          ; RESET
007162 004737 004500 JSR  %7,CPRDY          ; CHECK FOR PUNCH READY.
007166 013700 001206 MOV  PPB,%0
007172 005200          INC  %0
007174 112710 000000 MOVB #0,@%0          ; LOAD PPB+1
007200 105777 172000 TSTB @PPS          ; TEST PPS
007204 100401          BMI          .+4          ; BRANCH IF READY BIT NOT RESET.
007206 104015          ERROR          ; ERROR. LOADING PPB+1 CLEARED READY BIT.
007210 104000          SCOPE          ; SCOPE
007212          TSTA  PIH,1000.
007212          TSTAA PIH,1000.,\X+1,\X+2,\Y

;*****
007212 000007          PIT7: 7          ; PRG1 ROUTINE 7          *
007214 007260          PIT10         ; ADDRESS OF NEXT ROUTINE *
007216 001750          1000.          ; TEST ITERATION COUNT   *
007220 007226          PIHA          ; SCOPE ENTRY POINT      *
          000007          X=X+1

```

```

;*****
;TEST THAT PUNCH (READY BIT) IS ABLE TO INTERRUPT. IF THE INTERRUPT IS
;SERVICED, IT WILL HAVE OCCURRED AT CORRECT VECTOR.
007222 104014 STPCHV ;SET UP PUNCH INTERRUPT VECTOR.
007224 007256 PIHB
007226 005037 177776 PIHA: CLR PSW ;SET PRTY TO 0.
007232 004737 004500 JSR %7,CPRDY ;CHECK FOR PUNCH READY.
007236 042777 000100 171740 BIC #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
007244 052777 000100 171732 BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT
007252 000240 NOP
007254 104015 ERROR ;ERROR. FAILURE TO INTERRUPT WITH
;PUNCH READY BIT SET.
;INTERRUPT VECTOR POINTS HERE.

007256 104000 PIHB: SCOPE
007260 TSTA PII,1000.
007260 TSTAA PII,1000.,\X+1,\X+2,\Y
;*****
PIT10: 10 ; PRG1 ROUTINE 10 *
007262 007352 PIT11 ; ADDRESS OF NEXT ROUTINE *
007264 001750 1000. ; TEST ITERATION COUNT *
007266 007270 PIIA ; SCOPE ENTRY POINT *
000010 X=X+1
;*****
;TEST THAT PUNCH DOES NOT REINTERRUPT AFTER RTI WHEN READY BIT IS NOT RESET.
007270 104014 STPCHV ;SET UP PUNCH INTERRUPT VECTOR
007272 007326 PIIB
007274 005037 177776 CLR PSW ;SET PRTY TO 0.
007300 004737 004500 JSR %7,CPRDY ;CHECK FOR PUNCH READY.
007304 042777 000100 171672 BIC #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
007312 052777 000100 171664 BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT
007320 000240 NOP
007322 104015 ERROR ;ERROR. PUNCH FAILED TO INTERRUPT.
007324 104000 SCOPE ;SCOPE
007326 012777 007346 171660 PIIB: MOV #PIID,@PCHVTR ;CHANGE INTERRUPT VECTOR TO PIID
007334 012716 007342 MOV #PIIC,@%6 ;CHANGE INTERRUPT RETURN ADDRESS.
007340 000002 RTI ;RETURN FROM INTERRUPT.
007342 000240 PIIC: NOP
007344 104000 SCOPE
007346 104015 PIID: ERROR ;ERROR2. PUNCH REINTERRUPTED AFTER RTI WITH
007350 104000 SCOPE ;READY BIT LEFT ON. POP THE STOCK TWICE
007352 TSTA PIJ,1000.
007352 TSTAA PIJ,1000.,\X+1,\X+2,\Y
;*****
PIT11: 11 ; PRG1 ROUTINE 11 *
007354 007426 PIT12 ; ADDRESS OF NEXT ROUTINE *
007356 001750 1000. ; TEST ITERATION COUNT *
007360 007366 PIJA ; SCOPE ENTRY POINT *
000011 X=X+1
;*****
;TEST THAT THE PUNCH DOES NOT INTERRUPT WITH PROCESSOR AT SAME PRIORITY
;LEVEL AS THE PUNCH.
007362 104014 STPCHV ;SET UP PUNCH INTERRUPT VECTOR.
007364 007422 PIJB
007366 013737 001216 177776 PIJA: MOV PCHLVL,PSW ;SET PROCESSOR PRIORITY SAME AS PUNCH.
007374 005077 171604 CLR @PPS ;DISABLE PUNCH INTERRUPT.
007400 004737 004500 JSR %7,CPRDY ;CHECK FOR PUNCH READY.
007404 052777 000100 171572 BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT.

```

```

007412 000240
007414 005077 171564
007420 104000
007422 104015
                                NUP
                                CLR @PPS
                                SCOPE
                                ERROR
                                ;OK IF NO INTERRUPT OCCURS.
                                ;DISABLE PUNCH INTERRUPT.
                                ;SCOPE
                                ;ERROR. PUNCH ERRONEOUSLY INTERRUPTED
                                ;WITH PROCESSOR AT SAME PRIORITY LEVEL
                                ;AS THE PUNCH, OR THE PUNCH IS AT HIGHER
                                ;PRIORITY LEVEL THAN SPECIFIED AT PCHLVL.

007424 104000
007426
007426
                                SCOPE
                                TSTA P1K,1000.
                                TSTAA P1K,1000.,\X+1,\X+2,\Y
                                ;*****
PIT12: 12 ; PRG1 ROUTINE 12 *
        PIT13 ; ADDRESS OF NEXT ROUTINE *
        1000. ; TEST ITERATION COUNT *
        P1KA ; SCOPE ENTRY POINT *
        X=X+1
                                ;*****
                                ;TEST THAT PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY 1 LEVEL LOWER
                                ;THAN THE PUNCH PRIORITY.
                                STPCHV ;SET UP PUNCH INTERRUPT VECTOR
                                P1KB
                                P1KA: MOV PCHLVL,PSW ;SET PROCESSOR PRIORITY ONE LEVEL LOWER
                                SUB #40,PSW ;THAN PUNCH PRIORITY.
                                BIC #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
                                JSR %7,CPRDY ;CHECK FOR PUNCH READY.
                                BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT.
                                NOP
                                ERROR ;PUNCH FAILED TO INTERRUPT WITH PROCESSOR
                                ;PRIORITY ONE LEVEL LOWER THAN PUNCH.
                                ;THEREFORE, PUNCH PRIORITY MUST
                                ;BE LOWER THAN SPECIFIED AT PCHLVL.
                                ;HERE IF INTERRUPT OCCURS.

007436 104014
007440 007502
007442 013737 001216 177776
007450 162737 000040 177776
007456 042777 000100 171520
007464 004737 004500
007470 052777 000100 171506
007476 000240
007500 104015
                                P1KB: SCOPE
                                TSTA P1L,1000.
                                TSTAA P1L,1000.,\X+1,\X+2,\Y
                                ;*****
PIT13: 13 ; PRG1 ROUTINE 13 *
        PIT14 ; ADDRESS OF NEXT ROUTINE *
        1000. ; TEST ITERATION COUNT *
        P1LA ; SCOPE ENTRY POINT *
        X=X+1
                                ;*****
                                ;TEST THAT PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING CP PRIORITY TO 0.
                                ;SET UP PUNCH INTERRUPT VECTOR
                                STPCHV
                                P1LB
                                P1LA: JSR %7,CPRDY ;CHECK FOR PUNCH READY.
                                BIC #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
                                BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT
                                CLR PSW ;LOWER PRTY TO 0.
                                MOV #PRTY7,PSW ;RAISE CP PRIORITY BACK TO 7.
                                ERROR ;ERROR. PUNCH FAILED TO INTERRUPT IMMEDIATELY
                                ;AFTER CP PRIORITY WAS LOWERED TO 0.
                                ;HERE IF INTERRUPT OCCURS.

007502 104000
007504
007504
                                P1LB: SCOPE
                                TSTA P1M,5
                                TSTAA P1M,5,\X+1,\X+2,\Y
                                ;*****
PIT14: 14 ; PRG1 ROUTINE 14 *
        PIT15 ; ADDRESS OF NEXT ROUTINE *
    
```

```

007562 000005          5          ;TEST ITERATION COUNT          *
007564 007566          P1MA          ;SCOPE ENTRY POINT          *
          000014          X=X+1
;*****
;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 0 (FULL SPEED)
007566 012737 001000 001330 P1MA: MOV #512,RCNT ;SET CHARACTER COUNT TO 512
007574 004537 010062 JSR %5,PFRT ;GO PUNCH FRONT END AND MODE 0
007600 000000          0          ;INDICATOR
007602 104026          INBIN          ;INITIALIZE SPECIAL BINARY COUNT
007604 104030          GETBNP          ;GET BINARY CHARACTER
007606 004737 004532 JSR %7,HSPCH ;GO PUNCH THE CHARACTER
007612 005337 001330 DEC RCNT ;DECREMENT CHAR COUNT.
007616 001372          BNE P1MB ;BRANCH IF COUNT NOT YET 0 YET.
007620 104000          SCOPE          ;SCOPE
007622          TSTA PIN,5
007622          TSTAA PIN,5,\X+1,\X+2,\Y
;*****
PIT15: 15          ;PRG1 ROUTINE 15          *
          PIT16          ;ADDRESS OF NEXT ROUTINE          *
          5          ;TEST ITERATION COUNT          *
          P1NA          ;SCOPE ENTRY POINT          *
          X=X+1
;*****
;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 1 (RANDOM STALLS AFTER
;PUNCHING EACH CHARACTER. MAXIMUM STALL 47 MILLISECONDS)
007632 012737 177720 004202 P1NA: MOV #177720,STLMSK ;SET STALL MASK FOR 57(8) MAX
007640 012737 001000 001330 MOV #512,RCNT ;SET CHARACTER COUNT TO 512.
007646 004537 010062 JSR %5,PFRT ;GO PUNCH FRONT END, AND MODE 1
007652 000001          1          ;INDICATOR
007654 104026          INBIN          ;INITIALIZE SPECIAL BINARY COUNT.
007656 104030          GETBNP          ;GET BINARY CHARACTER.
007660 004737 004532 JSR %7,HSPCH ;GO PUNCH THE CHARACTER.
007664 104012          STALL          ;RANDOM STALL.
007666 005337 001330 DEC RCNT ;DECREMENT CHAR COUNT.
007672 001371          BNE PINB ;BRANCH IF COUNT NOT YET 0.
007674 104000          SCOPE          ;SCOPE
007676          TSTA P10,5
007676          TSTAA P10,5,\X+1,\X+2,\Y
;*****
PIT16: 16          ;PRG1 ROUTINE 16          *
          PIT17          ;ADDRESS OF NEXT ROUTINE          *
          5          ;TEST ITERATION COUNT          *
          P10A          ;SCOPE ENTRY POINT          *
          X=X+1
;*****
;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 2.
;(RANDOM STALL BEFORE PUNCHING RANDOM LENGTH GROUP OF CHARACTERS).
;MAXIMUM STALL 47 MILLISECONDS, MAXIMUM GROUP LENGTH -15)
007706 012737 177720 004202 P10A: MOV #177720,STLMSK ;SET STALL MASK FOR 57(8) MAX.
007714 012737 177760 004230 MOV #177760,RCMSK ;SET CHAR GROUP MASK FOR 17(8) MAX).
007722 012737 001000 001330 P10A: MOV #512,RCNT ;SET CHARACTER COUNT TO 512.
007730 004537 010062 JSR %5,PFRT ;GO PUNCH FRONT END AND MODE 2
007734 000002          2          ;INDICATOR
007736 104026          INBIN          ;INITIALIZE SPECIAL BINARY COUNT.
007740 004737 004204 P10B: JSR %7,GRCNT ;GENERATE RANDOM CHARACTER COUNT
007744 104012          STALL          ;RANDOM STALL.

```

```

007746 104030          P10C:  GETBNP          ;GET BINARY CHARACTER.
007750 004737 004532   JSR      %7,HSPCH      ;PUNCH THE CHARACTER.
007754 005337 001330   DEC      RCNT          ;DECREMENT CHAR COUNT
007760 001404          BEQ      P10D          ;BRANCH IF COUNT IS 0.
007762 005337 004232   DEC      RNCNT        ;NOT 0. DECREMENT RANDOM CHAR COUNT.
007766 001367          BNE      P10C          ;BRANCH IF COUNT NOT YET 0.
007770 000763          BR       P10B          ;BRANCH IF COUNT 0.
007772 104000          P10D:  SCOPE          ;SCOPE
007774          TSTAA   PIP,1,\X+1,LST,\Y

;*****
PIT17:  17             ; PRG1 ROUTINE 17 *
        PITLST        ; ADDRESS OF NEXT ROUTINE *
        1             ; TEST ITERATION COUNT *
        PIPA          ; SCOPE ENTRY POINT *
        X=X+1

;*****
;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 3.
;STALL 10 SECONDS, PUNCH 32 CHARACTERS, UNTIL THE ENTIRE PATTERN IS
;COMPLETED.
010004 012737 001000 001330 P1PA:  MOV      #512,RCNT      ;SET CHARACTER COUNT TO 512.
010012 004537 010062          JSR      %5,PFRT      ;GO PUNCH FRONT END AND MODE 3
010016 000003          3             ;INDICATOR.
010020 104026          INBIN                    ;INITIALIZE SPECIAL BIN COUNT
010022 104031          P1PB:  DELAY                    ;STALL 10 SECONDS
010024 023420          10000.
010026 012737 000040 004232 P1PC:  MOV      #32.,RNCNT      ;SET GROUP COUNT TO 32.
010034 104030          GETBNP          ;GET BINARY CHARACTER
010036 004737 004532   JSR      %7,HSPCH      ;PUNCH CHARACTER
010042 005337 001330   DEC      RCNT          ;DECREMENT CHAR COUNT
010046 001404          BEQ      P1PD          ;BRANCH IF COUNT IS 0
010050 005337 004232   DEC      RNCNT        ;DECREMENT GROUP COUNT
010054 001367          BNE      P1PC          ;BRANCH IF COUNT NOT YET 0.
010056 000761          BR       P1PB          ;BRANCH IF COUNT 0.
010060 104000          P1PD:  SCOPE          ;SCOPE

;SUBROUTINE TO PUNCH FRONT END AND MODE CODE (USED BY PRG3).
010062 012701 000024   PFRT:  MOV      #20,%1      ;PUNCH 20 BLANK CHARACTERS (000)
010066 005037 001332   CLR      PCHOUT        ;CLEAR PCHOUT.
010072 004737 004532   JSR      %7,HSPCH      ;PUNCH CHAR.
010076 005301          DEC      %1            ;DECREMENT R1
010100 001374          BNE      -6            ;BRANCH IF NOT 20 CHARCTERS YET.
010102 012737 000377 001332   MOV      #377,PCHOUT    ;PUNCH RUBOUT CHAR (SYNC CHAR).
010110 004737 004532   JSR      %7,HSPCH
010114 012537 001332   MOV      (5)+,PCHOUT    ;MOVE MODE CODE TO PCHOUT
010120 004737 004532   JSR      %7,HSPCH      ;PUNCH MODE CODE.
010124 012701 000004   MOV      #4,%1          ;PUNCH 4 BLANK CHARACTERS.
010130 005037 001332   CLR      PCHOUT
010134 004737 004532   JSR      %7,HSPCH
010140 005301          DEC      %1
010142 001374          BNE      -6
010144 000205          RTS      %5           ;EXIT

;PRG2 - PUNCH VERIFY PROGRAM
;THIS PROGRAM VERIFIES TAPE PRODUCED BY PRG1.
;ANY ERRORS FOUND ARE REPORTED.
010146 104010          PRG2:  TYPE           ;TYPE TITLE
    
```

010150	013454				TITL2		
010152	004737	004760			JSR	PC,RDRSEL	;SELECT READER.
010156	104011				TYPES		
010160	013507				IM2		
010162	013203				IM6		
010164	014001				IM23		
010166	177777				-1		
010170	104024				CHALT		;WAIT FOR USER.
010172	012737	000372	001352	ETOA:	MOV	#250,CTRA	;250 TO CTRA.(TOTAL CHAR COUNT).
010200	012737	000012	001354	ETOB:	MOV	#10,CTRB	
010206	004737	004024		ETOC:	JSR	%7,BREAD	;READ CHAR
010212	005737	001334			TST	CRBUF	
010216	001007				BNE	ETOD	;BRANCH IF NON-ZERO CHAR.
010220	005337	001354			DEC	CTRB	;0 CHAR. DECREMENT CTRB
010224	001413				BEG	ETOF	;BRANCH IF 10 CONSECUTIVE 0'S READ.
010226	005337	001352			DEC	CTRA	;NO. DECREMENT CTRA.
010232	001365				BNE	ETOC	;BRANCH IF NOT YET 250 CHARS READ.
010234	000403				BR	ETOE	;250 CHARS READ. SYNE ERROR.
010236	005337	001352		ETOD:	DEC	CTRA	;DECREMENT CTRA
010242	001356				BNE	ETOB	;BRANCH IF NOT 250 CHARS READ YET.
010244	104016			ETOE:	ERRORN		;SYNC ERROR. 250 CHARS READ WITHOUT
010246	015160				EM3		;A SUCCESSFUL SYNC.
010250	177777				-1		
010252	000747				BR	ETOA	;GO TRY AGAIN.
010254	004737	004024		ETOF:	JSR	%7,BREAD	;READ CHAR
010260	005737	001334			TST	CRBUF	
010264	001004				BNE	ETOG	;BRANCH IF NON-ZERO CHAR.
010266	005337	001352			DEC	CTRA	;DECREMENT CTRA
010272	001370				BNE	ETOF	;BRANCH IF NOT 250 CHARS READ YET.
010274	000763				BR	ETOE	;250 CHARS READ. SYNC ERROR.
010276	012737	000377	001326	ETOG:	MOV	#377,ERRT	
010304	043737	001350	001326		BIC	RDRMSK,ERRT	
010312	023737	001326	001334		CMP	ERRT,CRBUF	;COMPARE CHAR READ TO 377.
010320	001414				BEG	ETOH	;377.OK.
010322	104017				OACNV		;ERROR TYPEOUT.
010324	001326				ERRT		
010326	015222				ESB		
010330	000004				4		
010332	104017				OACNV		
010334	001334				CRBUF		
010336	015235				EWAS		
010340	000004				4		
010342	104016				ERRORN		;LEADER ERROR. SHOULD BE 377.
010344	015175				EM4		
010346	177777				-1		
010350	000710				BR	ETOA	;START OVER
010352	004737	004024		ETOH:	JSR	%7,BREAD	;READ CHAR.
010356	023727	001334	000003		CMP	CRBUF,#3	;COMPARE CHAR READ TO 3.
010364	101410				BLOS	ETOI	;BRANCH IF SAME OR LOWER.
010366	104017				OACNV		;ERROR. CONVERT DATA READ TO ASCII.
010370	001334				CRBUF		;SET UP FOR TYPEOUT.
010372	015314				FWAS		
010374	000004				4		
010376	104016				ERRORN		;LEADER ERROR. SHOULD BE BETWEEN
010400	015242				EM5		;0 AND 3.
010402	177777				-1		


```

010404 000672
010406 012737 000004 001352 ETOI: BR ETOA ;START OVER.
010414 005037 003152 CLR MOV #4,CTRA ;4 TO CTRA (CHAR COUNT)
010420 004737 004024 ETOJ: CLR BINR ;CLEAR BINR. EXPECTED CHAR IS 0.
010424 004737 010470 JSR %7,BREAD ;READ CHAR.
010430 005337 001352 JSR %7,ECHK ;GO CHECK CHAR READ.
010434 001371 DEC CTRA ;DECREMENT CTRA
010436 104026 BNE ETOJ ;BRANCH IF NOT 4 CHARS READ YET.
010440 012737 001000 001352 INBIN ;INITIALIZE SPECIAL BINARY COUNT.
010446 004737 004024 ETOK: MOV #512,CTRA ;SET CHAR COUNT TO 512.
010452 104027 JSR %7,BREAD ;READ CHAR.
010454 004737 010470 GETBNR ;GET BIN CHAR.
010460 005337 001352 JSR %7,ECHK ;GO CHECK CHAR READ.
010464 001370 DEC CTRA ;DECREMENT CHAR COUNT
010466 000641 BNE ETOK ;BRANCH IF NOT 512 CHARS READ YET.
010470 023737 001334 003152 ECHK: BR ETOA ;DONE. START OVER.
010476 001413 CMP CRBUF,BINR ;COMPARE CHAR READ AGAINST EXPECTED CHAR.
010500 104017 BEQ ECHKA ;BRANCH IF EQUAL.
010502 003152 OACNV ;CONVERT EXPECTED DATA TO ASCII.
010504 015066 BINR
010506 000004 ASB
010510 104017 4 ;
010512 001334 OACNV ;CONVERT DATA READ TO ASCII.
010514 015101 CRBUF
010516 000004 AWAS
010520 104016 4 ;
010522 015043 ERRORN ;ERROR. DATA ERROR.
010524 177777 EM1
010526 000207 -1
ECHKA: RTS %7 ;EXIT
;PRG3 - COMBINED READER-PUNCH TEST
;USES SPECIAL BINARY COUNT PATTERN.
PRG3: TYPE ;TYPE TITLE.
TITL3
JSR PC,RDRSEL ;SELECT READER
JSR PC,PCHSEL ;SELECT PUNCH.
TYPES
IM3
IM6
IM23
-1
CHALT
INEIN ;INITIALIZE BINARY COUNTS.
MOV #177620,STLMSK ;SET MAX. STALL DELAY.
CLR PCHCNT ;CLEAR PUNCH COUNT
CLR RBUSY ;CLEAR READER BUSY INDICATOR
STDRV ;SET PTRI VECTOR.
WZERO
STPCHV ;SET PTPI VECTOR.
PBIN
GETBNP
MOV BINP,PPPB ;GET BIN CHARACTER
MOV #BIT6,PPPS ;PUNCH IT
BIS ;ENABLE PTPI
CLR PSW ;SET PRIORITY 0.
TRAP ;TRAP CALL TO CAUSE NOISE.
BR -2
PBIN: TST PPS ;TEST FOR ERROR.
    
```

010642	100004			BPL	PBNA					; BRANCH IF NO ERROR.
010644	104010			TYPE						; TYPE PUNCH NOT READY
010646	014647			SM3						; MESSAGE.
010650	104024			CHALT						
010652	000771			BR	PBIN					; RECHECK FOR ERROR.
010654	105777	170324		TSTB	PPS					; CHECK FOR DONE.
010660	100403			BMI	PBNB					; BRANCH IF DONE SET.
010662	104016			ERRORN						; ERROR.FALSE PUNCH INTERRUPT.
010664	015411			EM11						
010666	177777			-1						
010670	005237	011064		INC	PCHCNT					; INCREMENT PUNCH COUNT.
010674	104030			GETBNP.						; GET BINARY CHARACTER
010676	013777	003154	170302	MOV	BINP, PPS					; ENABLE PUNCH
010704	023727	011064	000144	CMP	PCHCNT, #100.					; NOT BUSY. PUNCH COUNT 100 YET?
010712	103001			BHIS	.+4					; BRANCH IF YES.
010714	000002			RTI						; NOT YET. EXIT INTERRUPT
010716	105737	011066		TSTB	RBUSY					; READER BUSY?
010722	100406			BMI	PBNC					; BR IF YES.
010724	052737	000200	011066	BIS	#BIT7, RBUSY					; SET READER BUSY
010732	052777	000101	170240	BIS	#101, PPS					; ENABLE PTRI AND READER.
010740	023727	011064	000146	CMP	PCHCNT, #102.					; PUNCH COUNT 102?
010746	101402			BLOS	PBND					; BR IF NOT.
010750	005077	170230		CLR	PPS					; STOP PUNCH.
010754	104012			PBND:	STALL					
010756	000002			RTI						; EXIT INTERRUPT.
010760	005777	170214		CREAD:	TST	PPRS				; CHECK FOR ERROR.
010764	100003			BPL	CRDA					; BRANCH IF NO ERROR.
010766	004737	003732		JSR	%7, TSM2					; ERROR.TYPE MESSAGE.
010772	000772			BR	CREAD					
010774	105777	170200		CRDA:	TSTB	PPRS				; TEST FOR DONE.
011000	100403			BMI	CRDAA					; BRANCH IF DONE SET.
011002	104016			ERRORN						; ERROR. FALSE READER INTERRUPT.
011004	015361			EM10						
011006	177777			-1						
011010	017737	170166	001334	CRDAA:	MOV	PPRS, CRBUF				; CHARACTER READ TO CRBUF
011016	005337	011064		DEC	PCHCNT					
011022	001015			BNE	CREADC					; BR IF NOT 0.
011024	032777	000100	170152	BIT	#BIT6, PPS					; NO. PTPI ENABLED?
011032	001003			BNE	CREADA					; BR IF YES.
011034	052777	000100	170142	BIS	#BIT6, PPS					; NO. ENABLE PTPI.
011042	042737	000200	011066	CREADA:	BIC	#BIT7, RBUSY				; YES. CLEAR READER BUSY.
011050	005077	170124		CLR	PPRS					; DISABLE PTRI.
011054	000207			RTS	%7					; EXIT.
011056	005277	170116		CREADC:	INC	PPRS				; ENABLE READER
011062	000207			RTS	%7					; EXIT.
011064	000000			PCHCNT:	OPEN					
011066	000000			RBUSY:	OPEN					
011070	004737	010760		WZERO:	JSR	%7, CREAD				; READ CHARACTER
011074	005737	001334		TST	CRBUF					; CHECK CHARACTER
011100	001001			BNE	.+4					; BRANCH IF NON-ZERO CHAR.
011102	000002			RTI						; ZERO. EXIT INTERRUPT.
011104	012777	011122	170076	MOV	#RBN, PDRVTR					; SET READER VECTOR TO READ BINARY
011112	012737	000003	001344	MOV	#3, ERCTR					; COUNT. SET ERROR COUNTER TO 3.
011120	000402			BR	RBN					
011122	004737	010760		RBN:	JSR	%7, CREAD				; READ CHARACTER.
011126	104027			RBN:	GETBNR					; GET BINARY CHARACTER

```

011130 023737 003152 001334      CMP      BINR,CRBUF      ;COMPARE AGAINST CHAR READ.
011136 001001                      BNE      RBINB        ;BRANCH IF NOT SAME.
011140 000002                      RTI                          ;SAME EXIT INTERRUPT.
011142 104017                      RBINB: OACNV          ;CONVERT EXPECTED CHAR TO ASCII
011144 003152                      BINR
011146 015066                      ASB
011150 000004                      4
011152 104017                      OACNV          ;CONVERT RECEIVED CHAR TO ASCII
011154 001334                      CRBUF
011156 015101                      AWAS
011160 000004                      4
011162 104016                      ERRORN        ;ERROR MESSAGE. DATA ERROR.
011164 015043                      EM1
011166 177777                      -1
011170 005337 001344              DEC      ERCTR      ;3RD ERROR?
011174 001001                      BNE      RBINC      ;YES.
011176 000002                      RTI                          ;NO. EXIT INTERRUPT.
011200 052737 100000 011066      RBINC: BIS      #BIT15,RBUSY ;DISABLE STALLS.
011206 012777 011230 167774      MOV      #RBIND,DRDRVTR ;SET PTR VECTOR TO RBIND.
011214 012737 000004 001344      MOV      #4,ERCTR    ;USE ERCTR AS CHARACTER COUNTER.
011222 012700 001334              MOV      #CRBUF,%0   ;ADDR OF CRBUF TO %0
011226 000002                      RTI                          ;EXIT INTERRUPT
011230 004737 010760              RBIND: JSR      %7,CREAD ;READ CHARACTER
011234 013720 001334              MOV      CRBUF,(0)+ ;STORE CHARACTER STARTING AT CHR1
011240 005337 001344              DEC      ERCTR      ;3RD CHARACTER?
011244 001401                      BEQ      .+4          ;YES.
011246 000002                      RTI                          ;EXIT INTERRUPT. NOT 3RD YET.
011250 004737 004400              JSR      %7,SYNCA    ;GO SYNC THE READER.
011254 000751                      BR      RBINC        ;NO SYNC. TRY AGAIN.
011256 012777 011122 167724      MOV      #RBIN,DRDRVTR ;SYNCED. SET READER VECTOR TO RBIN.
011264 012737 000003 001344      MOV      #3,ERCTR    ;SET ERROR COUNT TO 3.
011272 042737 100000 011066      BIC      #BIT15,RBUSY ;ENABLE STALLS.
011300 000002                      RTI                          ;EXIT INTERRUPT.

:PRG4 - PUNCHES 2 CHARACTERS SET IN SR.
PRG4: TYPE
      TITL4
      JSR      PC,PCHSEL ;SELECT PUNCH.
      TYPES
      IM4
      IM16
      IM23
      -1
      CHALT
011302 104010                      ;COMMON HALT. WAIT FOR USER.
011304 013560                      ;PUNCH FIRST CHARACTER.
011306 004737 004562              MOV      SR,PCHOUT
011312 104011                      JSR      %7,HSPCH
011314 013623                      MOV      SR+1,PCHOUT ;PUNCH SECOND CHARACTER.
011316 013254                      JSR      %7,HSPCH
011320 014001                      BR      PRG4A
011322 177777                      ;REPEAT.
011324 104024                      ;COMMON HALT. WAIT FOR USER.
011326 113737 177570 001332      PRG4A: MOV      SR,PCHOUT ;PUNCH FIRST CHARACTER.
011334 004737 004532              JSR      %7,HSPCH
011340 113737 177571 001332      MOV      SR+1,PCHOUT ;PUNCH SECOND CHARACTER.
011346 004737 004532              JSR      %7,HSPCH
011352 000765                      BR      PRG4A
;PRG5 - READS-CHECKS TAPE PUNCHED WITH CODES SET IN SR
PRG5: TYPE
      TITL5
      JSR      PC,RDRSEL ;SELECT READER.
      TYPES
      IM5
      IM6
      IM23

```

011374	177777				-1						
011376	104024				CHALT						
011400	113737	177570	001356		MOV B	SR,CTRC					;COMMON HALT. WAIT FOR USER.
011406	113737	177571	001360		MOV B	SR+1,CTRD					;STORE EXPECTED CHARACTERS.
011414	004737	004024			JSR	%7,BREAD					;IN CTRC AND CTRD.
011420	013737	001334	001336	HTOA:	MOV	CRBUF,CHR1					;MATCH CHARS ON TAPE AGAINST EXPECTED CHARS.
011426	004737	004024			JSR	%7,BREAD					;READ CHAR INTO CHR1
011432	013737	001334	001340		MOV	CRBUF,CHR2					;READ CHAR
011440	023737	001336	001356		CMP	CHR1,CTRC					;INTO CHR2
011446	001041				BNE	HTOE					; (CHR1)=(CTRC)?
011450	023737	001340	001360		CMP	CHR2,CTRD					;NO.
011456	001062				BNE	HTOG					;YES. (CHR2)=(CTRD)?
011460	005037	001354			CLR	CTRB					;NO. MATCH ERROR.
011464	012737	000003	001344	HTOB:	MOV	#3,ERCTR					;YES. NEXT CHAR SHOULD = (CTRC) (CTRB=0)
011472	004737	004024		HTOC:	JSR	%7,BREAD					;3 TO ERROR COUNTER.
011476	005137	001354			COM	CTRB					;READ CHAR
011502	001437				BEQ	HTOF					;COMPLEMENT CHAR INDICATOR
011504	023737	001334	001356		CMP	CRBUF,CTRC					;BRANCH IF EXPECTED CHAR SHOULD = (CTRD)
011512	001767				BEQ	HTOC					; (CRBUF) = (CTRC)?
011514	104017				OACNV						;YES.
011516	001356				CTRC						;NO. (CTRC) TO ASB IN ASCII FORM.
011520	015066				ASB						
011522	000004				4						
011524	104017			HTOD:	OACNV						; (CRBUF) TO AWAS IN ASCII FORM.
011526	001334				CRBUF						
011530	015101				AWAS						
011532	000004				4						
011534	104016				ERRORN						;ERROR 1 CALL. TYPE DATA ERROR MESSAGE.
011536	015043				EM1						
011540	177777				-1						
011542	005337	001344			DEC	ERCTR					;3 ERRORS?
011546	001722				BEQ	HTOA					;YES. START ALL OVER.
011550	000750				BR	HTOC					;NO. CONTINUE READING.
011552	023737	001336	001360	HTOE:	CMP	CHR1,CTRD					; (CHR1) = (CTRD)?
011560	001021				BNE	HTOG					;NO. MATCH ERROR.
011562	023737	001340	001356		CMP	CHR2,CTRC					;YES. (CHR2) = (CTRC)?
011570	001015				BNE	HTOG					;NO. MATCH ERROR.
011572	012737	177777	001354		MOV	#-1,CTRB					;YES. NEXT CHAR SHOULD = (CTRD)
011600	000731				BR	HTOB					;GO START READING.
011602	023737	001334	001360	HTOF:	CMP	CRBUF,CTRD					; (CRBUF) = (CTRD)?
011610	001730				BEQ	HTOC					;YES. OK. CONTINUE READING.
011612	104017				OACNV						;NO. (CTRD) TO ASB IN ASCII FORM.
011614	001360				CTRD						
011616	015066				ASB						
011620	000004				4						
011622	000740				BR	HTOD					;GO GENERATE ERROR MESSAGE.
011624	104016			HTOG:	ERRORN						;MATCH ERROR. UNABLE TO MATCH UP
011626	015321				EM6						;2 CONSECUTIVE CHARACTERS FROM READER
011630	177777				-1						
011632	000670				BR	HTOA					;TO CHARACTERS READ FROM SR.
					:PRG6 -	READ X CHARACTERS, STALL Y MILLISECONDS.					
011634	104010			PRG6:	TYPE						;TYPE TITLE AND INSTRUCTIONS.
011636	013277				TITL6						
011640	004737	004760			JSR	PC,RDRSEL					;SELECT READER.
011644	104010				TYPE						

```

011646 013331          IM17
011650 104024          CHALT
011652 005037 011714  ITA:  CLR  ITY
011655 005037 011720          CLR  ITX
011662 113737 177570 011714  MOVB SR,ITY          ;MOVE STALL COUNT TO ITY.
011670 113737 177571 011720  MOVB SR+1,ITX        ;MOVE CHAR COUNT TO ITX.
011676 001405          BEQ  ITC          ;BR IF COUNT 0.
011700 004737 003654  ITB:  JSR  %7,AREAD      ;FETCH CHARACTER.
011704 105337 011720          DECB ITX          ;DECREMENT CHAR COUNT.
011710 001373          BNE  ITB          ;BRANCH IF COUNT NOT 0.
011712 104031          ITC:  DELAY          ;READ CHARS. STALL NOW.
011714 000000          ITY:  OPEN          ;STALL COUNT IN MSECS.
011716 000755          BR   ITA          ;REPEAT
011720 000000          ITX:  OPEN
;PRG7. PUNCH SPECIAL BINARY COUNT PATTERN TEST TAPE
PRG7:  TYPE          ;TYPE TITLE.
      TITL7
      JSR  PC,PCHSEL ;SELECT PUNCH.
      TYPES
      IM16
      -1
      CHALT          ;WAIT FOR USER.
011740 104024          MOV  #20,-(6)      ;PUNCH 20 BLANK CHAR. LEADER
011742 012746 000024  PRG7A: JSR  %7,HSPCH
011746 005037 001332  DEC  @%6
011752 004737 004532  BNE  PRG7A
011756 005316          INBIN          ;INITIALIZE SPECIAL BINARY COUNT
011760 001374          PRG7B: GETBNP      ;GET BINARY CHARACTER.
011762 104026          JSR  %7,HSPCH      ;PUNCH CHARACTER
011764 104030          BR   PRG7B      ;REPEAT.
011766 004737 004532  ;PRG10 - READER SPEED PRINT LOOP
011772 000774          PRG10: TYPE          ;TYPE TITLE
      TITL10
      JSR  PC,RDRSEL ;SELECT READER.
      TYPES
      IM10
      IM24A
      -1
      CHALT          ;HALT. WAIT FOR USER.
012014 104024          CLR  CTRC          ;CLEAR WORK REGISTERS
012016 005037 001356  KTA:  CLR  CTRB
012022 005037 001354          BIT  #BIT14,SR
012026 032737 040000 177570  BEQ  KTB          ;DETERMINE WHETHER 30 OR
012034 001403          MOV  #270.,CTRC ;300 SECOND TIMING IS DESIRED
012036 012737 000416 001356  KTB:  ADD  #30.,CTRC ;SET UP FOR DESIRED TIME BASE.
012044 062737 000036 001356  BR   KTD
012052 000407          JSR  %7,BREAD      ;READ CHARACTER.
012054 004737 004024  KTC:  DEC  CTRA      ;DECREMENT CTRA
012060 005337 001352          BNE  KTE          ;BRANCH IF CTRA NOT 0.
012064 001005          INC  CTRB          ;CTRA0.+1 TO CTRB.
012066 005237 001354          MOV  CTRC,CTRA      ;RELOAD CTRA.
012072 013737 001356 001352  KTD:  TST  SR          ;TIME UP?
012100 005737 177570  KTE:  BPL  KTC          ;NO.
012104 100363          JSR  %5,CPKPL      ;GO TYPE OUT DEVICE SPEED.
012106 004537 012214  KTF:  SM4
012112 014672

```

```

012114 000737          BR      KTA-2          ;GO HALT.
          :PRG11 - PUNCH SPEED PRINT LOOP ;TYPE TITLE.
          PRG11: TYPE
          TITL11
          JSR      PC,PCHSEL          ;SELECT PUNCH.
          TYPES
          IM16
          IM24A
          -1
          CHALT          ;HALT. WAIT FOR USER.
          LTA: CLR      CTRB          ;CLEAR WORK AREAS.
          CLR      PCHOUT
          BR      LTC
          LTB: JSR      %7,HSPCH          ;PUNCH A 0
          DEC      CTRA          ;DECREMENT CTRA
          BNE      LTD          ;BRANCH IF CTRA NOT 0
          INC      CTRB          ;INCREMENT CTRB.
          LTC: MOV      #60.,CTRA          ;MOVE 60 TO CTRA
          LTD: TST      SR          ;TIME UP?
          BPL      LTB          ;NO.
          LTE: JSR      %5,CPKPL          ;GO TYPE OUT DEVICE SPEED.
          SMS
          BR      LTA-2          ;GO HALT AND READY UP FOR NEXT TIME.
          CPKPL: MOV      (5)+,CPKPLA          ;MOVE ADDR OF 1ST MESSAGE TO CPKPLA.
          BDCNV          ;CONVERT (CTRB) TO DECIMAL ASCII.
          CTRB
          ACPS
          3
          CPKPLA: TYPES          ;TYPE DEVICE SPEED.
          OPEN
          ACPS
          -1
          RTS      %5          ;EXIT.
          ;
  
```

```

:PA611 ADDITIONAL CODE TO CHECK PUNCH LOGIC AND READER LIGHTS

```

```

:MODIFIED: MARCH 15, 1975
:PROGRAMMER: MIKE MITCHELL

```

```

:PROBLEM CORRECTED: PUNCH GETS HUNG WHEN RUNNING OUT OF TAPE
                    AND OPERATOR TURNS PUNCH OFF. THIS PREVENTS
                    THE PUNCH FROM FINISHING THE PUNCH CYCLE
                    BY KEEPING READY LOW.

```

```

:PROBLEM FIX: ADDITIONAL LOGIC ALLOWS PUNCH TO BE
              RE-INITED UNDER PROGRAM CONTROL.

```

```

:THIS CODE RUNS UNDER OPERATOR INTERVENTION.

```

```

012242 177777
012244 000000

```

```

MONE: 177777 ;BINARY ONE PATTERN
ZERO: 0 ;ZERO PATTERN

```

```

:PROGRAM 12---HIGH SPEED PUNCH PROGRAMMABLE INIT TEST

```

```

012246 104010
012250 015645
012252 004737 004562
012256 104011
012260 013254
012262 014001
012264 177777
012266 104024

```

```

PRG12: TYPE
       TITL12 ;TEXT "%PRG12--PROGRAMMABLE INIT TEST%"
       JSR PC,PCHSEL ;SELECT THE PUNCH
       TYPES
       IM16 ;TEXT "MAKE PUNCH READY"
       IM23 ;TEXT "HIT CONTINUE"
       -1
       CHALT ;WAIT FOR OPERATOR.

```

```

012270 042777 000006 166706
012276 004737 012404
012302 113737 012242 001332
012310 004737 012332
012314 113737 012244 001332
012322 004737 012332
012326 000137 012302

```

```

PG12B: BIC #6,PPPS ;DISABLE PUNCH INTERRUPT.
        JSR PC,TSTPUN ;CHECK FOR PUNCH READY.
PG12A: MOVB MONE,PCHOUT ;PUNCH BINARY 1 PATTERN.
        JSR PC,HSPCH1 ;DO IT HERE.
        MOVB ZERO,PCHOUT ;PUNCH BINARY ZERO PATTERN.
        JSR PC,HSPCH1
        JMP PG12A ;LOOP

```

```

:HIGH SPEED PUNCH ROUTINE

```

```

012332 000240
012334 043737 001346 001332
012342 013777 001332 166636
012350 004737 012436
012354 000000
012356 052777 000400 166620
012364 032737 040000 177570
012372 001371

```

```

HSPCH1: NOP ;DEBUG.
        ;PUNCH THE CHARACTER
        BIC PCHMSK,PCHOUT
        MOV PCHOUT,PPPB ;LOAD PUNCH BUFFER.
        JSR PC,TIMER ;WAIT FOR DONE
        HALT ;IN A TIMEOUT LOOP.
        BIS #400,PPPS ;TIMED-OUT RETURN
        BIT #BIT14,SR ;RE-INIT PUNCH.
        BNE IS ;SCOPE LOOP ENABLED???
        ;YES,LOOP UNTIL SR14=0.

```

012374 004737 012436
012400 000000JSR PC, TIMER
HALT;RE-CHECK THE DONE BIT.
;FATAL ERROR: DONE BIT WAS NO
;RESET BY THE INIT INSTRUCTION.

;ELSE RETURN IS TO THE NEXT INSTRUCTION.

012402 000207

;OUTPUT RTS PC ;EXIT
;OUTPUT ERROR MESSAGE OF PUNCH NOT READY
;IF ERROR BIT 15 SET OR BIT 7 NOT SET.012404 005777 166574
012410 100404
012412 105777 166566
012416 100001
012420 000207
012422 104011
012424 014647
012426 013254
012430 177777
012432 104024
012434 000763TSTPUN: TST @PPS
BMI HSPCH2
TSTB @PPS
BPL HSPCH2
RTS PC
HSPCH2: TYPES
SM3
IM16
-1
CHALT
BR TSTPUN;"PUNCH NOT READY"
;"MAKE PUNCH READY"
;TERMINATOR

;LOOP AGAIN

;TIMER ROUTINE FOR HIGH SPEED PUNCH

;WAITS FOR READY FLAG IN LOOP
;IF READY COMES UP WITHIN ALLOWABLE TIME ; RETURN IS TO CALL+2
;ELSE INIT IS ISSUED TO HSP AND PROGRAM HALTS AT CALL+1012436 012737 177200 001354
012444 000240
012446 005237 001354
012452 001406
012454 105777 166524
012460 100405TIMER: MOV #-600,CTRB ;MAXIMUM=600 MS DELAY
TIMER1: NOP
INC CTRB ;UP COUNTER
BEQ TIMER2 ;TIMED-OUT...ERROR..
TSTB @PPS ;READY BIT SET?
BMI TIMER3 ;YES, OK

;READY NOT SET, SO DELAY 10 MS AND CHECK AGAIN.

012462 104031
012464 000012
012466 000766DELAY
10.
BR TIMER1 ;10 MILLI SE DELAY.
;GO CHECK AGAIN.012470 000240
012472 000207TIMER2: NOP
RTS PC ;RETURN TO CALL+1012474 062716 000002
012500 000207TIMER3: ADD #2,(SP) ;RETURN TO CALL+2 IF NO ERROR
RTS PC;ADDITIONAL CODE TO PROVIDE READER LIGHT TEST UNDER OPERATOR
;INTERVENTION.
;USE SWITCH 8 TO TURN READER LIGHT OFF.
;USE SWITCH 12(1) TO SELECT NEW READER TO TEST.

012502 104010
 012504 015713
 012506 004737 004760
 012512 104011
 012514 015745
 012516 016026
 012520 177777

012522 013700 177570
 012526 032700 000400
 012532 001773
 012534 052777 000400 166436
 012542 032737 040000 177570
 012550 001371
 012552 000000
 012554 032737 010000 177570
 012562 001351
 012564 000752

PRG13: TYPE
 TITL13
 PRG13C: JSR PC,RDRSEL
 PRG13B: TYPES
 IM13A
 IM13B
 -1
 :SWB=1 TO TURN LIGHT OFF.
 PG13A: MOV SR,RO
 BIT #400,RO
 BEQ PG13A
 1\$: BIS #400,APRS
 BIT #BIT14,SR
 BNE 1\$
 HALT
 BIT #10000,SR
 BNE PRG13C
 BR PRG13B

;OUTPUT TEST HEADER.
 ;TEXT "PRG13--READER LIGHT TEST"
 ;SELECT READER VIA SWR.
 ;OUTPUT MESSAGE:
 ;TEXT "TURN LIGHT ON VIA RDR SWITCH"
 ;TEXT "USE SWB TO TURN LIGHT
 ; OFF"

;CHECK BIT 8.
 ;SWITCH DOWN, NO ACTION REQUIRED.
 ;ELSE TURN LIGHT OFF.
 ; IS SCOPE LOOP SET?
 ;YES, LOOP UNTIL SRI4=0.
 ;WAIT FOR OPR TO HIT CONTINUE
 ;CHECK FOR SELECT NEW READER.
 ;YES, SELECT NEW READER.

N05

PA611 MACY11 27(1006) 04-NOV-76 12:14 PAGE 65
DZPAAA.CMB 04-NOV-76 12:11

012566 022445 120

APGEND: .ASCII '%P'

012571	040	020040	047105	APN:	.ASCII	' END.0'
012576	027104	100				
012601	045	037445	047111	CM2:	.ASCII	'%%?INVALID PROGRAM0'
012606	040526	044514	020104			
012614	051120	043517	040522			
012622	040115					
012624	022445	044477	053116	CM3:	.ASCII	'%%?INVALID TEST0'
012632	046101	042111	052040			
012640	051505	040124				
012644	022445	042523	020124	ASETSR:	.ASCII	'%%SET SR OPTIONS. NORMAL SR IS 0000000'
012652	051123	047440	052120			
012660	047511	051516	020056			
012666	047516	046522	046101			
012674	051440	020122	051511			
012702	030040	030060	030060			
012710	040060					
012712	022445	054524	042520	PGTIT:	.ASCII	'%%TYPESET 11 READER-PUNCH TESTS%0'
012720	042523	020124	030461			
012726	051040	040505	042504			
012734	026522	052520	041516			
012742	020110	042524	052123			
012750	022523	100				
012753	045	042522	052123	RUNINS:	.ASCII	'%RESTART PROGRAM.0'
012760	051101	020124	051120			
012766	043517	040522	027115			
012774	100					
012775	045	042523	020124	MSVCTR:	.ASCII	'%SET SR WITH RDR0 VECTOR.0'
013002	051123	053440	052111			
013010	020110	042122	030122			
013016	053040	041505	047524			
013024	027122	100				
013027	045	042523	020124	SELRDR:	.ASCII	'%SET # OF READERS IN SR.0'
013034	020043	043117	051040			
013042	040505	042504	051522			
013050	044440	020116	051123			
013056	040056					
013060	051445	052105	021440	SELPCH:	.ASCII	'%SET # OF PUNCHES IN SR.0'
013066	047440	020106	052520			
013074	041516	042510	020123			
013102	047111	051440	027122			
013110	100					
013111	045	050045	043522	TITL1:	.ASCII	'%%PRG1: PUNCH TEST.0'
013116	027061	050040	047125			
013124	044103	052040	051505			
013132	027124	100				
013135	045	050045	043522	TITL7:	.ASCII	'%%PRG7. COUNT PATTERN TAPE GENERATOR.0'
013142	027067	041440	052517			
013150	052116	050040	052101			
013156	042524	047122	052040			
013164	050101	020105	042507			
013172	042516	040522	047524			
013200	027122	100				
013203	045	040515	042513	IM6:	.ASCII	'%MAKE READER READY.0'
013210	051040	040505	042504			
013216	020122	042522	042101			
013224	027131	100				

013227	045	050045	043522	TITLQ:	.ASCII	'%PRGO. READER TEST.0'
013234	027060	051040	040505			
013242	042504	020122	042524			
013250	052123	040056				
013254	046445	045501	020105	IM16:	.ASCII	'%MAKE PUNCH READY.0'
013262	052520	041516	020110			
013270	042522	042101	027131			
013276	100					
013277	045	050045	043522	TITL6:	.ASCII	'%PRG6 - READ X, STALL Y.0'
013304	020066	020055	042522			
013312	042101	054040	020054			
013320	052123	046101	020114			
013326	027131	100				
013331	045	042523	020124	IM17:	.ASCII	'%SET SR15 TO SR8 TO NO. OF CHARS TO READ,'
013336	051123	032461	052040			
013344	020117	051123	020070			
013352	047524	047040	027117			
013360	047440	020106	044103			
013366	051101	020123	047524			
013374	051040	040505	026104			
013402	051445	052105	051440		.ASCII	'%SET SR7 TO SR0 TO NO. OF MSECs TO STALL.0'
013410	033522	052040	020117			
013416	051123	020060	047524			
013424	047040	027117	047440			
013432	020106	051515	041505			
013440	020123	047524	051440			
013446	040524	046114	040056			
013454	022445	051120	031107	TITL2:	.ASCII	'%PRG2. PUNCH VERIFY TEST.0'
013462	020056	052520	041516			
013470	020110	042526	044522			
013476	054506	052040	051505			
013504	027124	100				
013507	045	047514	042101	IM2:	.ASCII	'%LOAD READER WITH TAPE PRODUCED '
013514	051040	040505	042504			
013522	020122	044527	044124			
013530	052040	050101	020105			
013536	051120	042117	041525			
013544	042105	040				
013547	102	020131	051120		.ASCII	'BY PRG1.0'
013554	030507	040056				
013560	022445	051120	032107	TITL4:	.ASCII	'%PRG4. PUNCHES 2 CODES SET IN SR.0'
013566	020056	052520	041516			
013574	042510	020123	020062			
013602	047503	042504	020123			
013610	042523	020124	047111			
013616	051440	027122	100			
013623	045	042523	020124	IM4:	.ASCII	'%SET CODES TO BE PUNCHED IN SR.0'
013630	047503	042504	020123			
013636	047524	041040	020105			
013644	052520	041516	042510			
013652	020104	047111	051440			
013660	027122	100				
013663	045	050045	043522	TITL5:	.ASCII	'%PRG5. READS TAPE PUNCHED WITH CODES SET IN SR.0'
013670	027065	051040	040505			
013676	051504	052040	050101			
013704	020105	052520	041516			

013712	042510	020104	044527	
013720	044124	041440	042117	
013726	051505	051440	052105	
013734	044440	020116	051123	
013742	040056			
013744	051445	052105	041440	IMS: .ASCII '%SET CODES TO BE READ IN SR.3'
013752	042117	051505	052040	
013760	020117	042502	051040	
013766	040505	020104	047111	
013774	051440	027122	100	
014001	040	051120	051505	IM23: .ASCII 'PRESS CONTINUE.3'
014006	020123	047503	052116	
014014	047111	042525	040056	
014022	022445	051120	030507	TITL10: .ASCII '%PRG10. RDR SPEED TEST.3'
014030	027060	051040	051104	
014036	051440	042520	042105	
014044	052040	051505	027124	
014052	100			
014053	045	047514	042101	IM10: .ASCII '%LOAD ANY TAPE LOOP IN READER '
014060	040440	054516	052040	
014066	050101	020105	047514	
014074	050117	044440	020116	
014102	042522	042101	051105	
014110	040			
014111	101	042116	046440	.ASCII 'AND MAKE READY.'
014116	045501	020105	042522	
014124	042101	027131		

014130	051445	052105	051440
014136	030522	020064	047524
014144	040440	030440	043040
014152	051117	031440	030060
014160	051440	041505	040
014165	124	046511	047111
014172	026107	047440	020122
014200	042523	020124	052111
014206	052040	020117	020060
014214	047506	020122	030063
014222	040		

.ASCII '%SET SRI4 TO A 1 FOR 300 SEC '

.ASCII 'TIMING, OR SET IT TO 0 FOR 30 '

F06

PA611 MACY11 27(1006) 04-NOV-76 12:14 PAGE 70
DZPAAA.CMB 04-NOV-76 12:11

014223	123	041505	047117
014230	020104	044524	044515
014236	043516	040056	

.ASCII 'SECOND TIMING.2'

014242	050045	042522	051523	IM24A:	.ASCII	'%PRESS CONTINUE TO START TIMING.'
014250	041440	047117	044524			
014256	052516	020105	047524			
014264	051440	040524	052122			
014272	052040	046511	047111			
014300	027107					
014302	051445	052105	051440		.ASCII	'%SET SR 15 TO A 1 AT END OF '
014310	020122	032461	052040			
014316	020117	020101	020061			
014324	052101	042440	042116			
014332	047440	020106				
014336	044524	044515	043516		.ASCII	'TIMING PERIOD TO OBTAIN DEVICE SPEED '
014344	050040	051105	047511			
014352	020104	047524	047440			
014360	052102	044501	020116			
014366	042504	044526	042503			
014374	051440	042520	042105			
014402	040					
014403	124	050131	047505		.ASCII	'TYPEOUT.3'
014410	052125	040056				
014414	022445	051120	030507	TITL11:	.ASCII	'%PRG11. PCH SPEED TEST.3'
014422	027061	050040	044103			
014430	051440	042520	042105			
014436	052040	051505	027124			
014444	100					
014445	045	050045	043522	TITL3:	.ASCII	'%PRG3. COMBINED READER-PUNCH TEST.3'
014452	027063	041440	046517			
014460	044502	042516	020104			
014466	042522	042101	051105			
014474	050055	047125	044103			
014502	052040	051505	027124			
014510	100					
014511	045	040515	042513	IM3:	.ASCII	'%MAKE PUNCH READY, PUNCH BLANK LEADER, '
014516	050040	047125	044103			
014524	051040	040505	054504			
014532	020054	052520	041516			
014540	020110	046102	047101			
014546	020113	042514	042101			
014554	051105	020054				
014560	047514	042101	051040		.ASCII	'LOAD READER.3'
014566	040505	042504	027122			
014574	100					
014575	045	042522	042101	SM1:	.ASCII	'%READER ERROR BIT SET.3'
014602	051105	042440	051122			
014610	051117	041040	052111			
014616	051440	052105	040056			
014624	051045	040505	042504	SM2:	.ASCII	'%READER NOT READY.3'
014632	020122	047516	020124			
014640	042522	042101	027131			
014646	100					

014647	045	050045	047125	SM3:	.ASCII	'%PUNCH NOT READY.␣'
014654	044103	047040	052117			
014662	051040	040505	054504			
014670	040056					
014672	022445	042522	042101	SM4:	.ASCII	'%READER SPEED : ␣'
014700	051105	051440	042520			
014706	042105	035040	040040			
014714	022445	052520	041516	SMS:	.ASCII	'%PUNCH SPEED : ␣'
014722	020110	050123	042505			
014730	020104	020072	100			
014735	040	020040	020040	ACPS:	.ASCII	' CHARS PER SEC.␣'
014742	044103	051101	020123			
014750	042520	020122	042523			
014756	027103	100				
014761	045	050045		EMO:	.ASCII	'%P'
014764	020040	020040	124	APNUMB:	.ASCII	' T'
014771	040	020040	020040	ATNUMB:	.ASCII	' PC'
014776	041520					
015000	020040	020040	020040	APC:	.ASCII	' ICNT '
015006	020040	041511	052116			
015014	040					
015015	040	020040	020040	AICNT:	.ASCII	' .␣'
015022	040056					
015024	020040	050106	020103	FPCMSG:	.ASCII	' FPC '
015032	020040	020040	020040	AFPC:	.ASCII	' %␣'
015040	022440	100				
015043	040	042040	052101	EM1:	.ASCII	' DATA ERROR S/B: '
015050	020101	051105	047522			
015056	020122	051440	041057			
015064	020072					
015066	020040	020040	020040	ASB:	.ASCII	' WAS: '
015074	040527	035123	040			
015101	040	020040	040040	AWAS:	.ASCII	' ␣'
015106	051040	051105	040505	EM2:	.ASCII	' REREAD ERROR. 1ST READ: '
015114	020104	051105	047522			
015122	027122	020040	051461			
015130	020124	042522	042101			
015136	020072					
015140	020040	020040	020040	ORGRD:	.ASCII	' WAS: '
015146	040527	035123	040			
015153	040	020040	040040	SUBRD:	.ASCII	' ␣'
015160	051440	047131	020103	EM3:	.ASCII	' SYNC ERROR.␣'
015166	051105	047522	027122			
015174	100					
015175	045	042514	042101	EM4:	.ASCII	'%LEADER ERROR. S/B: '
015202	051105	042440	051122			
015210	051117	020056	051440			
015216	041057	020072				
015222	020040	020040	020040	ESB:	.ASCII	' WAS: '
015230	040527	035123	040			
015235	040	020040	040040	EWAS:	.ASCII	' ␣'
015242	046045	040505	042504	EMS:	.ASCII	'%LEADER ERROR. S/B BETWEEN '
015250	020122	051105	047522			
015256	027122	051440	041057			
015264	041040	052105	042527			
015272	047105	040				

015275	060	040440	042116	.ASCII	'0 AND 3. WAS :
015302	031440	020056	040527		
015310	020123	020072			
015314	020040	020040	100	FWAS:	.ASCII ' @'
015321	040	040515	041524	EM6:	.ASCII ' MATCH ERROR.@'
015326	020110	051105	047522		
015334	027122	100			
015337	045	047516	051040	EM7:	.ASCII '%NO RDR RESPONSE.@'
015344	051104	051040	051505		
015352	047520	051516	027105		
015360	100				
015361	040	040506	051514	EM10:	.ASCII ' FALSE READER INTERRUPT@'
015366	020105	042522	042101		
015374	051105	044440	052116		
015402	051105	052522	052120		
015410	100				
015411	040	040506	051514	EM11:	.ASCII ' FALSE PUNCH INTERRUPT@'
015416	020105	052520	041516		
015424	020110	047111	042524		
015432	051122	050125	040124		
015440	051445	052105	051040	SRDRM:	.ASCII '%SET RDR # IN SR. SET SR15 IF 6 LEVEL.@'
015446	051104	021440	044440		
015454	020116	051123	020056		
015462	042523	020124	051123		
015470	032461	044440	020106		
015476	020066	042514	042526		
015504	027114	100			
015507	045	042523	020124	SPCHM:	.ASCII '%SET PCH # IN SR. SET SR15 IF 6 LEVEL.@'
015514	041520	020110	020043		
015522	047111	051440	027122		
015530	051440	052105	051440		
015536	030522	020065	043111		
015544	033040	046040	053105		
015552	046105	040056			
015556	037445	047111	040526	INVRP:	.ASCII '%?INVALID RDR/PCH.@'
015564	044514	020104	042122		
015572	027522	041520	027110		
015600	100				
015601	045	042122	020122	RDRIDM:	.ASCII '%RDR '
015606	020040	051440	046105	ARDRID:	.ASCII ' SELECTED.@'
015614	041505	042524	027104		
015622	100				
015623	045	041520	020110	PCHIDM:	.ASCII '%PCH '
015630	020040	051440	046105	APCHID:	.ASCII ' SELECTED.@'
015636	041505	042524	027104		
015644	100				
015645	045	050045	043522	TITL12:	.ASCII '%%PRG12--PROGRAMMABLE INIT PUNCH TEST@'
015652	031061	026455	051120		
015660	043517	040522	046515		
015666	041101	042514	044440		
015674	044516	020124	052520		
015702	041516	020110	042524		
015710	052123	100			
015713	045	051120	030507	TITL13:	.ASCII '%%PRG13--READER LIGHT TEST@'
015720	026463	051055	040505		
015726	042504	020122	044514		

015734	044107	020124	042524
015742	052123	100	
015745	045	052045	051125
015752	020116	044514	044107
015760	020124	043117	051040
015766	040505	042504	020122
015774	047117	053040	040511
016002	051040	051104	041440
016010	047117	051124	046117
016016	051440	044527	041524
016024	040110		
016026	050045	052125	051440
016034	034127	052040	020117
016042	020061	047524	052040
016050	051125	020116	044514
016056	044107	020124	043117
016064	027106	100	
	001450		

IM13A: .ASCII '%TURN LIGHT OF READER ON VIA RDR CONTROL SWITCH@'

IM13B: .ASCII '%PUT SWB TO 1 TO TURN LIGHT OFF.@'

.END START

POSA	006330	2177	2183#
POSB	006362	2182	2190#
POTA	006400	2197	2202#
POTLST=	177777	1091#	2265
POTO	005226	1860	1868#
POT1	005256	1869	1883#
POT10	005614	.995	2013#
POT11	005654	2014	2031#
POT12	005756	2032	2062#
POT13	006020	2063	2081#
POT14	006070	2082	2101#
POT15	006144	2102	2125#
POT16	006222	2126	2148#
POT17	006314	2149	2174#
POT2	005306	1884	1898#
POT20	006364	2175	2194#
POT21	006412	2195	2208#
POT22	006450	2209	2225#
POT23	006522	2226	2244#
POT24	006600	2245	2264#
POT3	005362	1899	1920#
POT4	005416	1921	1936#
POT5	005450	1937	1954#
POT6	005502	1955	1971#
POT7	005552	1972	1994#
POVA	006434	2211	2218#
POVB	006472	2228	2234#
POVB	006502	2236#	2239
POXA	006552	2247	2255#
POXB	006560	2257#	2260
POYA	006630	2267	2276#
POYB	006636	2278#	2281
P1AA	006716	2297	2302#
P1AB	006724	2301	2304#
P1BA	006746	2312	2317#
P1BB	006754	2316	2319#
P1CA	006770	2327	2331#
P1CB	007012	2333	2336#
P1DA	007044	2347	2351#
P1EA	007100	2363	2368#
P1FA	007122	2378	2382#
P1GA	007160	2395	2399#
P1HA	007226	2414	2421#
P1HB	007256	2420	2428#
P1IA	007270	2435	2439#
P1IB	007326	2440	2448#
P1IC	007342	2449	2451#
P1ID	007346	2448	2453#
P1JA	007366	2461	2468#
P1JB	007422	2467	2475#
P1KA	007442	2485	2492#
P1KB	007502	2491	2502#
P1LA	007520	2509	2515#
P1LB	007554	2514	2522#
P1MA	007566	2529	2533#
P1MB	007604	2537#	2540

EMTDEF	1025#	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189
	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201			
TITLE	1025#														
TSTA	1025#	1865	1880	1895	1917	1933	1951	1968	1991	2010	2028	2059	2078	2098	2122
	2145	2171	2191	2205	2222	2241	2291	2306	2321	2341	2357	2372	2389	2408	2429
	2455	2479	2503	2523	2542	2564									
TSTAA	1025#	1866	1881	1896	1918	1934	1952	1969	1992	2011	2029	2060	2079	2099	2123
	2146	2172	2192	2206	2223	2242	2262	2292	2307	2322	2342	2358	2373	2390	2409
	2430	2456	2480	2504	2524	2543	2565	2592							

. ABS. 016067 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

.NOW.SEQ/SOL/CRF/NL:TOC=DZPAAA.CMB
RUN-TIME: 11 21 3 SECONDS
RUN-TIME RATIO: 111/36=3.0
CORE USED: 10K (19 PAGES)

