

[Frame 1]	[Frame 2]	[Frame 3]	[Frame 4]	[Frame 5]	[Frame 6]
[Frame 7]	[Frame 8]	[Frame 9]	[Frame 10]	[Frame 11]	[Frame 12]
[Frame 13]	[Frame 14]	[Frame 15]	[Frame 16]	[Frame 17]	[Frame 18]
[Frame 19]	[Frame 20]	[Frame 21]	[Frame 22]	[Frame 23]	[Frame 24]
[Frame 25]	[Frame 26]	[Frame 27]	[Frame 28]	[Frame 29]	[Frame 30]
[Frame 31]	[Frame 32]	[Frame 33]	[Frame 34]	[Frame 35]	[Frame 36]
[Frame 37]	[Frame 38]	[Frame 39]	[Frame 40]	[Frame 41]	[Frame 42]
[Frame 43]	[Frame 44]	[Frame 45]	[Frame 46]	[Frame 47]	[Frame 48]
[Frame 49]	[Frame 50]	[Frame 51]	[Frame 52]	[Frame 53]	[Frame 54]
[Frame 55]	[Frame 56]	[Frame 57]	[Frame 58]	[Frame 59]	[Frame 60]
[Frame 61]	[Frame 62]	[Frame 63]	[Frame 64]	[Frame 65]	[Frame 66]
[Frame 67]	[Frame 68]	[Frame 69]	[Frame 70]	[Frame 71]	[Frame 72]
[Frame 73]	[Frame 74]	[Frame 75]	[Frame 76]	[Frame 77]	[Frame 78]
[Frame 79]	[Frame 80]	[Frame 81]	[Frame 82]	[Frame 83]	[Frame 84]
[Frame 85]	[Frame 86]	[Frame 87]	[Frame 88]	[Frame 89]	[Frame 90]
[Frame 91]	[Frame 92]	[Frame 93]	[Frame 94]	[Frame 95]	[Frame 96]
[Frame 97]	[Frame 98]	[Frame 99]	[Frame 100]	[Frame 101]	[Frame 102]
[Frame 103]	[Frame 104]	[Frame 105]	[Frame 106]	[Frame 107]	[Frame 108]
[Frame 109]	[Frame 110]	[Frame 111]	[Frame 112]	[Frame 113]	[Frame 114]
[Frame 115]	[Frame 116]	[Frame 117]	[Frame 118]	[Frame 119]	[Frame 120]
[Frame 121]	[Frame 122]	[Frame 123]	[Frame 124]	[Frame 125]	[Frame 126]
[Frame 127]	[Frame 128]	[Frame 129]	[Frame 130]	[Frame 131]	[Frame 132]
[Frame 133]	[Frame 134]	[Frame 135]	[Frame 136]	[Frame 137]	[Frame 138]
[Frame 139]	[Frame 140]	[Frame 141]	[Frame 142]	[Frame 143]	[Frame 144]
[Frame 145]	[Frame 146]	[Frame 147]	[Frame 148]	[Frame 149]	[Frame 150]

IDENTIFICATION

SEQ 0001

PRODUCT CODE: AC-9006C-MC
PRODUCT NAME: CZPAACO TYP-11 READER-PUNCH TESTS
DATE: JAN 15, 1972
MAINTAINER: DIAGNOSTIC GROUP

THIS DIAGNOSTIC OBSOLETES MAINDEC - 11 - D2EA

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1972,1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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 FOLLOWING 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. REQUIREMENTS

2.1 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.

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.

4. USE PROCEDURE
-----4.1 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 000000. 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 POCEND. 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.

ONE ERROR-FREE PASS WILL TAKE APPROXIMATELY 10 MINUTES.

4.3 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.

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.

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.

4.11 PRG12 USE PROCEDURE

THIS PROGRAM IS INTENDED TO BE USED AS AN AID IN CHECKING

J 1

OUT THE ADDITIONAL LOGIC ADDED TO THE PUNCH CONTROLLER THAT
ALLOWS THE PROGRAMMER TO ISSUE 'PUNCH RESET' UNDER SOFTWARE
CONTROL.

SEQ 0009

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.

7. SWITCH REGISTER OPTIONS:

SR14=1 TO SCOPE LOOP ON THE PROGRAMMABLE INIT FUNCTION

4.12 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.

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 (PRG0,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 SRO 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 SRO 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 000776.

- 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.

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 OZZZZZ ICNT VVVVV.'

WHERE:

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

PC OZZZZZ 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 PRGO 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 PCOZZZZZ.
- 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.

(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 PRG0 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 IT IS 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.

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:

(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. DESCRIPTION8.1 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.

8.2 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.

8.4 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.

8.8 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

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:

I 2

SEQ 0021

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.

1
2
3
4
5
6
7
8
9

: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

.LIST SRC
: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 - SCOPF.
: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

:
: 0
:+2 ;UNASSIGNED TRAP
MACHER: HALT
:+2 ;SP OVERFLOW, BUS ERROR TRAP
HALT
:+2 ;RESERVED INSTRUCTION TRAP
HALT

```
000014 002376 TRCV: SV5S ;TRACE TRAP
000016 000340 PRTY7
000020 002426 IOTV: RS5S ;TRAP TO CALL IOX
000022 000340 PRTY7
000024 000026 .+2 ;POWER FAIL TRAP
000026 000000 HALT
000030 002176 EMTV: EMTINT ;EMT TRAP
000032 000340 PRTY7
000034 000036 TRPV: .+2 ;TRAP TRAP. SIMILAR TO EMT.
000036 000002 RTI ;EXIT TRAP CALL.
000036 000040 .+40
;LOCATIONS 40 THROUGH 776 ARE FILLED WITH .+2 AND HALT.
.NLIST ME
.LIST ME
;EQUATE STATEMENTS
SR=177570
PSW=177776
000774 177746 CACHE: 177746 ; FOR CONTROL REGISTER FOR 11-70
001200 SPBOT=1200
000240 NOP=240
000000 OPEN=0
100000 MANUAL=BIT15
100000 BIT15=100000
040000 BIT14=40000
020000 BIT13=20000
010000 BIT12=10000
004000 BIT11=4000
002000 BIT10=2000
001000 BIT9=1000
000400 BIT8=400
000200 BIT7=200
000100 BIT6=100
000040 BIT5=40
000020 BIT4=20
000010 BIT3=10
000004 BIT2=4
000002 BIT1=2
000001 BIT0=1
000000 R0=%0
000001 R1=%1
000002 R2=%2
000003 R3=%3
000004 R4=%4
000005 R5=%5
000006 R6=%6
000007 R7=%7
000007 PC=%7
005746 PUSH=005746
024646 PUSH2=024646
005726 POPSP=005726
022626 POPSP2=022626
000340 PRTY7=340
000300 PRTY6=300
000240 PRTY5=240
000200 PRTY4=200
000140 PRTY3=140
```


	000100		PRTY2=100	
	000040		PRTY1=40	
	000000		PRTY0=0	
	000007		BELL=007	
	177777		POTLST=-1	
	177777		P1TLST=-1	
	177777		TLAST=-1	
	000003		TRC=3	
	000040		I=40	
	100000		A=BIT15	
	040000		B=BIT14	
	020000		C=BIT13	
	000100		IE=BIT6	
	000000		EMTX=0	
	000003		SAV5S=3	
	000004		RST5S=4	
	000060		.=60	
000060	002510		KBSVC	;KEYBOARD VECTOR SETUP.
000062	000340		PRTY7	;STATUS UPON KEYBOARD INTERRUPT.
	000200		.=200	
000200	000137	001450	JMP START	;GO TO START OF PROGRAM.
	001000		.=1000	
	001002		NOP	
	001004	001252	CLR INITD	;RESET INIT SWITCH.
001010	000137	001450	JMP START	;GO TO START OF PROGRAM.
	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	RDRLVL:	PRTY4	;READER PRIORITY LEVEL
001214	000000	PCHVTR:	OPEN	;PUNCH INTERRUPT VECTOR
001216	000200	PCHLVL:	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	IPVTR:	64	;LSP INTERRUPT VECTOR
001242	000200	IPLVL:	PRTY4	;LSP PRIORITY LEVEL
001244	000000	FSTVCT:	OPEN	
001246	000000	RDRLIM:	OPEN	
001250	000000	PCHLIM:	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	005244	PRGTAB: PRG0	:PRG0 START ADDRESS
001300	006716	PRG1	:PRG1 START ADDRESS
001302	010206	PRG2	:PRG2 START ADDRESS
001304	010570	PRG3	:PRG3 START ADDRESS
001306	011342	PRG4	:PRG4 START ADDRESS
001310	011414	PRG5	:PRG5 START ADDRESS
001312	011674	PRG6	:PRG6 START ADDRESS
001314	011762	PRG7	:PRG7 START ADDRESS
001316	012034	PRG10	:PRG10 START ADDRESS
001320	012156	PRG11	:PRG11 START ADDRESS
		:OPERATOR INTERVENTION TESTS:	
001322	012306	PRG12	:PUNCH TEST (OPR INTV.)
001324	012542	PRG13	:READER LIGH TEST
			:(OPR INTV REQD.)
001326	000000	ERRT: OPEN	
001330	000000	RCNT: OPEN	:CHARACTER COUNT
001332	000000	PCHOUT: OPEN	
001334	000000	CRBUF: OPEN	:HOLDS ONE CHARACTER FROM READER.
001336	000000	CHR1: OPEN	
001340	000000	CHR2: OPEN	
001342	000000	CHR3: OPEN	
001344	000000	ERCTR: OPEN	
001346	000000	PCHMSK: OPEN	
001350	000000	RDRMSK: OPEN	
001352	000000	CTRA: OPEN	
001354	000000	CTRB: OPEN	
001356	000000	CTRC: OPEN	
001360	000000	CTRD: OPEN	
001362	000000	FPC: OPEN	
		.NLIST MC	
001364		FMTTAB:	
001364	002012	CHAINN	:POINTER FOR EMT CALL SCOPE
001366	002452	SRSETT	:POINTER FOR EMT CALL SRESET
001370	002216	SV03	:POINTER FOR EMT CALL SAV03
001372	002316	RS03	:POINTER FOR EMT CALL RST03
001374	002246	SV05	:POINTER FOR EMT CALL SAV05
001376	002346	RS05	:POINTER FOR EMT CALL RST05
001400	002236	SV05S	:POINTER FOR EMT CALL SAV05S
001402	002342	RS05S	:POINTER FOR EMT CALL RST05S
001404	002742	TYP	:POINTER FOR EMT CALL TYPE
001406	003034	TYPS	:POINTER FOR EMT CALL TYPES
001410	004212	STAL	:POINTER FOR EMT CALL STALL
001412	004004	STPTRV	:POINTER FOR EMT CALL STRDRV
001414	004034	STTPV	:POINTER FOR EMT CALL STPCHV
001416	002566	ERR	:POINTER FOR EMT CALL ERROR
001420	002576	ERRN	:POINTER FOR EMT CALL ERRORN
001422	003344	OACNVV	:POINTER FOR EMT CALL OACNV
001424	003436	BDCNVV	:POINTER FOR EMT CALL BDCNV
001426	003620	RNGEN	:POINTER FOR EMT CALL RNDNUM
001430	003676	INRNDN	:POINTER FOR EMT CALL INRND
001432	003414	BMOVV	:POINTER FOR EMT CALL BMOVE
001434	002542	CHLT	:POINTER FOR EMT CALL CHALT
001436	002554	EHLT	:POINTER FOR EMT CALL EHALT
001440	003130	INBINN	:POINTER FOR EMT CALL INBIN

```

001442 003274          GTBINR          ;POINTER FOR EMT CALL GETBNR
001444 003216          GTBINP          ;POINTER FOR EMT CALL GETBNP
001446 003064          DLYN           ;POINTER FOR EMT CALL DELAY
                                .LIST      MC
001450 012706 001200   START:  MOV      #SPBOT,R6      ;SET BOTTOM OF SP STACK.
001454 005737 001252   TST      INITD          ;SEE IF PROGRAM IS INITIALIZED.
001460 001041          BNE      STRTA         ;BR IF YES.
001462 104010          TYPE           ;TYPE TITLE.
001464 012752          PGTIT           ;TYPE INSTRUCTIONS TO SET RDR 0 VECTOR.
001466 104011          TYPES
001470 013035          MSVCTR
001472 014041          IM23
001474 177777          -1
001476 104024          CHALT
001500 013737 177570 001244  MOV      SR,FSTVCT      ;SAVE RDRO VECTOR.
001506 104011          TYPES          ;TYPE INSTRUCTIONS TO SET # OF READERS.
001510 013067          SELRDR
001512 014041          IM23
001514 177777          -1
001516 104024          CHALT
001520 113737 177570 001246  MOV      SR,RDRLIM     ;SAVE # OF READERS.
001526 104011          TYPES          ;TYPE INSTRUCTIONS TO SET # OF PUNCHES.
001530 013120          SELPCH
001532 014041          IM23
001534 177777          -1
001536 104024          CHALT
001540 113737 177570 001250  MOV      SR,PCHLIM     ;SAVE # OF PUNCHES.
001546 012737 177777 001252  MOV      #-1,INITD    ;INITIALIZATION COMPLETE.
001554 104010          TYPE           ;RESTART MESSAGE.
001556 013013          RUNINS
001560 104024          CHALT
001562 000776          BR
001564 013746 000004   STRTA:  MOV      @#4,-(SP) ;SAVE CONTENTS OF ADDRESS 4
001570 013746 000006   MOV      @#6,-(SP) ;SAVE CONTENTS OF ADDRESS 6
001574 012737 001612 000004  MOV      #CCHE,@#4 ;LOAD FOR TIMEOUT WITH NON CACHE PROCESSOR
001602 052737 000014 000774  BIS      #14,@#CACHE ;DISABLE CACHE
001610 000401          BR AROUND ; BRANCH AROUND FOR CACHE PROCESSORS
001612 022626   CCHE:  CMP      (SP)+,(SP)+ ; CLEAN UP THE STACK FOR NON CACHE PROC.
001614 012637 000006   AROUND: MOV      (SP)+,@#6 ;RESTORE CONTENTS OF 6
001620 012637 000004   MOV      (SP)+,@#4 ;RESTORE CONTENTS OF 4
001624 012737 000340 177776  MOV      #PRTY7,PSW ;SET PRIORITY 7
001632 004737 005174   JSR      PC,CLNUP ;GO DO CLEAN-UP
001636 013700 177570   MOV      SR,RO ;GET PROGRAM NUMBER.
001642 042700 177760   BIC      #177760,RO
001646 020027 000013   CMP      RO,#13 ;VALID PROGRAM NUMBER?
001652 101404          BLOS     STRTB ;BR IF YES.
001654 104010          TYPE           ;TYPE INCORRECT PROGRAM MESSAGE.
001656 012641          CM2
001660 104024          CHALT
001662 000740          BR      STRTA ;TRY AGAIN.
001664 010037 001254   STRTB: MOV      RO,PRGNUM ;SAVE PROGRAM NUMBER.
001670 006300          ASL      RO ;RO TIMES 2.
001672 104001          SRESET ;SYSTEM RESET.
001674 000170 001276   SRSET: JMP      @PRGTAB(0) ;GO TO SELECTED PROGRAM.
001700 104011          TYPES          ;TYPE SR OPTION MESSAGE.
001702 012704          ASETSR

```

001704	014041				IM23			
001706	177777				-1			
001710	104024				CHALT			:COMMON HALT.
001712	013737	001256	001264	GETRDY:	MOV	KSTART,NXTST		:ADDR OF 1ST ROUTINE TO NXTST
001720	012737	000340	177776	GTRDYX:	MOV	#PRTY7,PSW		:SET PRIORITY 7.
001726	012706	001200			MOV	#SPBOT,R6		:SET BOTTOM OF STACK.
001732	104001				SRESET			:ISSUE RESET.
001734	004737	002136		GTRDYA:	JSR	R7,FORWD		:ROLL FORWARD TO 'NEXT' ROUTINE.
001740	032737	001000	177570	GTRDYB:	BIT	#BIT9,SR		:CHECK SELECT ROUTINE SWITCH
001746	001002				BNE	GTRDYC		:BRANCH IF SELECT ROUTINE SWITCH IS SET.
001750	000177	177304		GORUN:	JMP	@CURTST		:GO RUN CURRENT ROUTINE.
001754	013700	177570		GTRDYC:	MOV	SR,R0		:(SR) TO R0
001760	042700	177600			BIC	#177600,R0		:MASK UNDESIRED BITS
001764	123700	001262			CMPB	RTNNO,R0		:COMPARE RTNNO TO (R0)
001770	001767				BEQ	GORUN		:BR IF ROUTINE FOUND.
001772	022737	177777	001264	GTRDYD:	CMP	#-1,NXTST		:NO. CHECK FOR LAST ROUTINE.
002000	001355				BNE	GTRDYA		:BRANCH IF NOT LAST ROUTINE.
002002	104010				TYPE			:TYPE INCORRECT RTN SELECTED.
002004	012664				CM3			
002006	104024				CHALT			:COMMON HALT.
002010	000740				BR	GETRDY		:START OVER.
002012	012706	001200		CHAINN:	MOV	#SPBOT,R6		:RESTORE STACK.
002016	005237	001266			INC	ICNT		:INCREMENT ITERATION COUNT.
002022	001002				BNE	CHNAC		:BR IF RESULT NOT 0.
002024	005137	001266			COM	ICNT		:RESULT 0. RESET ICNT TO -1.
002030	032737	040000	177570	CHNAC:	BIT	#BIT14,SR		:CHECK FOR SCOPE OPTION.
002036	001402				BEQ	CHNA		:BRANCH IF SCOPE SW NOT SET.
002040	000177	177226		CHNAB:	JMP	@SCOPTR		:RETURN TO ROUTINE.
002044	032737	004000	177570	CHNA:	BIT	#BIT11,SR		:TEST INHIBIT ITERATION SWITCH
002052	001003				BNE	CHNAA		:BRANCH IF INHIBIT ITERATION SW SET.
002054	005337	001270			DEC	ICTR		:DECFMENT ITERATION COUNT.
002060	001367				BNE	CHNAB		:BRANCH IF COUNT NOT 0.
002062	032737	002000	177570	CHNAA:	BIT	#BIT10,SR		:ROUTINE END HALT SW SET? (SR10)
002070	001403				BEQ	CHNB		:BRANCH IF NOT SET.
002072	013700	001262			MOV	RTNNO,R0		:TEST # TO R0.
002076	000000				HALT			:ROUTINE END HALT. TEST # IN LIGHTS.
002100	032737	001000	177570	CHNB:	BIT	#BIT9,SR		:CHECK SELECT ROUTINE SWITCH
002106	001301				BNE	GETRDY		:BRANCH IF SELECT RTN SW SET
002110	022737	177777	001264		CMP	#-1,NXTST		:LAST TEST?
002116	001300				BNE	GTRDYX		:BRANCH IF NOT LAST TEST.
002120	104017				OACNV			:CONVERT PROGRAM NUMBER TO ASCII.
002122	001254				PRGNUM			
002124	012631				APN			
002126	000002				2			
002130	104010				TYPE			:TYPE PROGRAM END BELL.
002132	012626				APGEND			
002134	000666				BR	GETRDY		:GO REPEAT PROGRAM.
002136	013705	001264		FORWD:	MOV	NXTST,R5		:ADDR OF NEXT ROUTINE TO R5.
002142	012537	001262			MOV	(5)+,RTNNO		:GET NEXT ROUTINE NUMBER.
002146	012537	001264			MOV	(5)+,NXTST		:GET ADDR OF NEXT 'NEXT' ROUTINE.
002152	012537	001270			MOV	(5)+,ICTR		:GET ITERATION COUNT.
002156	012537	001272			MOV	(5)+,SCOPTR		:GET SCOPE LOOP ENTRY POINTER.
002162	010537	001260		FORWDA:	MOV	R5,CURTST		:ADDR OF NOW CURRENT TEST TO CURTST.
002166	012737	000001	001266		MOV	#1,ICNT		:PRESET ICNT TO 1.
002174	000207				RTS	R7		:EXIT FORWD SUBROUTINE.

;EMT INTERPRETER ROUTINE.

```

002176 010046          EMTINT: MOV      R0,-(6)          ;PUSH R0.
002200 016600 000002      MOV      2(6),R0          ;GET EMT PC.
002204 014000          MOV      -(0),R0         ;GET EMT CALL.
002206 006300          ASL      R0              ;TIMES 2.
002210 016000 171364      MOV      EMTTAB-10000(0),R0 ;FORM EMT ROUTINE ADDR.
002214 000200          RTS      R0              ;GO TO EMT ROUTINE; RESTORE R0.

;SAVE REGS 0 TO 3 SUBROUTINE.
002216 012666 177766      SV03:  MOV      (6)+,-10.(6) ;MOVE PC UPSTACK.
002222 012666 177766      MOV      (6)+,-10.(6) ;MOVE STATUS UPSTACK.
002226 012737 000002 002302  MOV      #RTI,SV05C
002234 000415          BR       SV05B

;SUB TO SAVE REGS 0 TO 5 AND PLACE EMT PC IN R5.
002236 012737 000240 002302  SV05S: MOV      #NOP,SV05C
002244 000403          BR       SV05A

;SUB TO SAVE REGS 0 TO 5.
002246 012737 000002 002302  SV05:  MOV      #RTI,SV05C
002254 012666 177762      SV05A: MOV      (6)+,-14.(6) ;MOVE PC AND PSW UPSTACK.
002260 012666 177762      MOV      (6)+,-14.(6)
002264 010546          MOV      R5,-(6)
002266 010446          MOV      R4,-(6)
002270 010346          SV05B: MOV      R3,-(6)
002272 010246          MOV      R2,-(6)
002274 010146          MOV      R1,-(6)
002276 010046          MOV      R0,-(6)
002300 024646          PUSH2
002302 000002          SV05C: RTI              ;RTI OR NOP.
002304 016605 000020      MOV      16.(6),R5      ;EMT PC TO R5.
002310 010504          MOV      R5,R4
002312 005744          TST      -(4)
002314 000002          RTI              ;EXIT.

;RESTORE REGS 0 TO 3 SUBROUTINE.
002316 022626          RS03:  POPSP2
002320 012600          MOV      (6)+,R0        ;RESTORE REGS 0 TO 4.
002322 012601          MOV      (6)+,R1
002324 012602          MOV      (6)+,R2
002326 012603          MOV      (6)+,R3
002330 016646 177766      MOV      -10.(6),-(6) ;MOVE PC AND PSW DOWN STACK.
002334 016646 177766      MOV      -10.(6),-(6)
002340 000002          RTI              ;EXIT.

;SUB TO SET R5 IN EMT PC AND RESTORE REGS 0 TO 5.
002342 010566 000020      RS05S: MOV      R5,16.(6) ;SET EMT PC TO R5 CONTENTS.

;SUB TO RESTORE REGS 0 TO 5.
002346 022626          RS05:  POPSP2
002350 012600          MOV      (6)+,R0
002352 012601          MOV      (6)+,R1
002354 012602          MOV      (6)+,R2
002356 012603          MOV      (6)+,R3
002360 012604          MOV      (6)+,R4
002362 012605          MOV      (6)+,R5
002364 016646 177762      MOV      -14.(6),-(6) ;MOVE PC AND PSW DOWNSTACK.
002370 016646 177762      MOV      -14.(6),-(6)
002374 000002          RTI              ;EXIT.
002376 012666 177772      SV5S:  MOV      (6)+,-6(6) ;PC AND PSW UPSTACK.
002402 012666 177772      MOV      (6)+,-6(6)
002406 010546          MOV      R5,-(6) ;SAVE R5.
002410 010446          MOV      R4,-(6) ;SAVE R4.

```

```

002412 024646          PUSH2
002414 016605 000010  MOV      8.(6),R5      ;EMT PC TO R5.
002420 010504          MOV      R5,R4        ;EMT PC TO R4.
002422 005744          TST      -(4)
002424 000002          RTI
002426 010566 000010  RS5S:  MOV      R5,8.(6) ;EXIT EMT SUB.
002432 022626          POPSP2                ;R5 TO EMT PC.
002434 012604          MOV      (6)+,R4      ;RESTORE R4.
002436 012605          MOV      (6)+,R5      ;RESTORE R5.
002440 016646 177772  MOV      -6(6),-(6)
002444 016646 177772  MOV      -6(6),-(6)
002450 000002          RTI                  ;EXIT.
;ROUTINE TO ISSUE RESET AND ENABLE KEYBOARD INTERRUPTS.
002452 104004          SRSETT: SAVO5
002454 012700 052525  MOV      #52525,R0    ;DATA TO R0.
002460 005100          COM      R0          ;COMPLEMENT (R0).
002462 010037 002' 5  MOV      R0,SRSETT+4 ; (R0) TO SRSETT+4.
002466 000005          RESET
002470 104005          RSTO5                ;RESET. R0 IS DISPLAYED.
002472 005737 000042  TST      @#42         ;LOADED FROM DECTAPE?
002476 001403          BEQ      SRSETA      ;BR IF NOT.
002500 052777 000100 176516  BIS      #BIT6,@TKS   ;ENABLE KEYBOARD INTERRUPTS.
002506 000002          SRSETA: RTI          ;EXIT.
;KEYBOARD SERVICE ROUTINE.
002510 017727 176512 000000  KBSVC: MOV      @TKB,#0    ;READ KEYBOARD BUFFER.
002516 042737 000200 002514  BIC      #BIT7,KBSVC+4 ;CLEAR PARITY BIT.
002524 022737 000003 002514  CMP      #3,KBSVC+4   ;IS IT CTRL C?
002532 001401          BEQ      .+4         ;BR IF YES.
002534 000002          RTI                  ;NO. EXIT.
002536 013707 000042  MOV      @#42,PC      ;EXIT TO DECTAPE MONITOR.
;COMMON HALT ROUTINE
002542 104006          CHLT:  SAVO5S
002544 010400          MOV      R4,R0      ;DEVELOP ADDR OF CALLER.
002546 000000          HALT                ;HALT CALL ADDR IN DATA LIGHTS.
002550 104007          RSTO5S
002552 000002          RTI                  ;EXIT.
;CONDITIONAL ERROR HALT ROUTINE.
002554 005737 177570  EHLT:  TST      SR      ;CHECK FOR HALT ON ERROR.
002560 100001          BPL      EHLTA       ;BRANCH IF NO HALT DESIRED.
002562 000000          HALT
002564 000002          EHLTA: RTI          ;IN DATA LIGHTS.
002566 012737 000406 002716  ERR:    MOV      #406,ERRNB ;SET UP FOR SINGLE MESSAGE.
002574 000403          BR      ERRN+6
002576 012737 000240 002716  ERRN:  MOV      #NOP,ERRNB ;SET UP FOR MULTIPLE MESSAGES.
002604 010437 001362  MOV      R4,FPC      ;CONVERT CALL ADDR OF SUB CALLING.
002610 104017          OACNV
002612 001362          FPC
002614 015072          AFPC
002616 000006          6
002620 000003          SAV5S
002622 010537 002714  MOV      R5,ERRB     ;SAVE REG 5S
002626 162737 000002 002714  SUB      #2,ERRB     ;DETERMINE CALLING ADDR.
002634 104017          OACNV
002636 002714          ERB
002640 015040          APC
002642 000006          6

```

```

002644 104017 OACNV ;CONVERT PROGRAM # TO ASCII.
002646 001254 PRGNUM
002650 015024 APNUMB
002652 000002 2
002654 104017 OACNV ;CONVERT TEST # TO ASCII.
002656 001262 RTNNO
002660 015031 ATNUMB
002662 000003 3
002664 104020 BDCNV ;CONVERT ICNT TO DECIMAL ASCII.
002666 001266 ICNT
002670 015055 AICNT
002672 000005 5
002674 012737 015021 002714 MOV #EMO,ERRB ;TYPE ERR HEADER MSG IF NOT INHIBITED.
002702 032737 020000 177570 ERRNA: BIT #BIT13,SR ;INHIBIT ERR PRINT?
002710 001002 BNE ERRNB ;BR TO INHIBIT.
002712 104010 TYPE ;TYPE MSG.
002714 000000 ERRB: OPEN ;DESIRED MSG ADDR GOES HERE.
002716 000000 ERRNB: OPEN ;NOP OR 406
002720 012537 002714 MOV (5)+,ERRB ;GET ADDR OF NEXT MSG.
002724 022737 177777 002714 CMP #-1,ERRB ;TERMINATOR?
002732 001363 BNE ERRNA ;GO TYPE IF NOT TERMINATOR.
002734 104025 ERRNC: EHALT ;END OF MSGS. HALT IF REQUIRED.
002736 000004 RST5S ;RESTORE REG 5S.
002740 000002 RTI ;EXIT EMT SUB.

;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
002742 104006 TYP: SAV5S
002744 012500 MOV (5)+,R0 ;ADDRESS OF MESSAGE TO R0.
002746 112001 TYP A: MOV B (0)+,R1 ;GET CHARACTER
002750 122701 000100 CMP B #100,R1 ;CHECK FOR 'a' CHARACTER
002754 001002 BNE TYP C ;BRANCH IF NOT 'a'.
002756 104007 RST5S
002760 000002 RTI ;TERMINATOR CHAR. DONE. EXIT.
002762 122701 000045 TYP C: CMP B #45,R1 ;CHECK FOR 'Z'.
002766 001411 BEQ TYP F ;BRANCH IF 'Z'.
002770 004737 002776 JSR R7,TYP D ;TYPE CHAR IN R1
002774 000764 BR TYP A
002776 110177 176230 TYP D: MOV B R1,@TPB ;OUTPUT CHARACTER TO PRINTER
003002 105777 176222 TST B @TPS ;WAIT FOR DONE FLAG.
003006 100375 BPL .-4
003010 000207 RTS R7 ;EXIT
003012 112701 000015 TYP F: MOV B #15,R1 ;MOVE CARRIAGE RETURN CODE TO R1
003016 004737 002776 JSR R7,TYP D ;GO TYPE CHAR.
003022 112701 000012 TYP G: MOV B #12,R1 ;MOVE LF CODE TO R1.
003026 004737 002776 JSR R7,TYP D ;GO TYPE CHAR.
003032 000745 BR TYP A

;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
003034 000003 TYP S: SAV5S
003036 012537 003060 MOV (5)+,TYP SB ;ADDR OF MESSAGE TO TYP SB.
003042 022737 177777 003060 CMP #-1,TYP SB ;CHECK FOR TERMINATOR
003050 001002 BNE TYP SA ;BRANCH IF NOT TERMINATOR.
003052 000004 RST5S
003054 000002 RTI ;TERMINATOR, EXIT
003056 104010 TYP SA: TYPE ;CALL ON TYP SUB TO TYPE MESSAGE
003060 000000 TYP SB: OPEN ;ADDRESS OF MESSAGE GOES HERE
003062 000765 BR TYP S+2 ;GO PROCESS NEXT MESSAGE

;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS

```

```

003064 011637 003124      DLYN:  MOV      (6),DLCNT      ;GET EMT PC.
003070 062716 000002      ADD      #2,(6)              ;SET UP EXIT PC.
003074 104002              SAV03                       ;SAVE REGS
003076 017700 000022      MOV      @DLCNT,R0          ;DELAY COUNT TO R0.
003102 001406              BEQ      DLYCN              ;BR IF 0.
003104 012701 000303      DLYAN:  MOV      #303,R1     ;1 MSEC COUNT TO R1.
003110 005301      DLYBN:  DEC      R1          ;DECREMENT 1 MSEC COUNT.
003112 001376              BNE      DLYBN             ;BR IF NOT 0.
003114 005300              DEC      R0                ;DECREMENT DELAY COUNT.
003116 001372              BNE      DLYAN             ;BR IF NOT DONE DELAYING.
003120 104003      DLYCN:  RST03
003122 000002              RTI                          ;EXIT.
003124 000000      DLCNT:  OPEN
003126 000000      DLCTR:  OPEN
;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
003130 012737 177777 003176  INBINN:  MOV      #-1,RIND     ;SET ALL VARIABLES
003136 013737 003176 003200      MOV      RIND,PIND
003144 013737 003176 003202      MOV      RIND,PT0
003152 013737 003176 003204      MOV      RIND,PT1
003160 013737 003176 003206      MOV      RIND,RT0
003166 013737 003176 003210      MOV      RIND,RT1
003174 000002              RTI                          ;EXIT.
003176 000000      RIND:  OPEN
003200 000000      PIND:  OPEN
003202 000000      PT0:  OPEN
003204 000000      PT1:  OPEN
003206 000000      RT0:  OPEN
003210 000000      RT1:  OPEN
003212 000000      BINR:  OPEN
003214 000000      BINP:  OPEN
;SPECIAL BINARY COUNT PATTERN SUBROUTINE (PUNCH)
003216 013737 003202 003204  GTBINP:  MOV      PT0,PT1     ;PREVIOUS BIN CHAR TO PT1
003224 005137 003204              COM      PT1
003230 005137 003200              COM      PIND
003234 001002              BNE      .+6
003236 005237 003204              INC      PT1
003242 043737 001346 003204  BIC      PCHMSK,PT1
003250 013737 003204 003202      MOV      PT1,PT0          ;SAVE BIN CHAR IN PT0
003256 013737 003204 003214      MOV      PT1,BINP         ;BIN CHAR TO BINP.
003264 013737 003204 001332      MOV      PT1,PCHOUT
003272 000002              RTI                          ;EXIT.
;SPECIAL BINARY COUNT PATTERN SUBROUTINE (READER)
003274 013737 003206 003210  GTBINR:  MOV      RT0,RT1     ;PREVIOUS BIN CHAR TO RT1.
003302 005137 003210              COM      RT1
003306 005137 003176              COM      RIND
003312 001002              BNE      .+6
003314 005237 003210              INC      RT1
003320 043737 001350 003210  BIC      RDRMSK,RT1
003326 013737 003210 003206      MOV      RT1,RT0          ;SAVE BIN CHAR IN RT0.
003334 013737 003210 003212      MOV      RT1,BINR         ;BIN CHAR TO BINR.
003342 000002              RTI                          ;EXIT.
;EMT SUB TO CONVERT OCTAL TO ASCII.
003344 104006      OACNVV: SAV05S             ;SAVE REGS.
003346 013500              MOV      @ (5)+,R0        ;GET OCTAL VALUE.
003350 012501              MOV      (5)+,R1         ;GET DESTINATION ADDR.
003352 012502              MOV      (5)+,R2         ;GET CONVERT COUNT.
    
```



```

003354 060201          ADD      R2,R1          ;DEVELOP ADDR TO STORE 1ST CHAR.
003356 010003          MOV      R0,R3
OACNVA: BIC      #177770,R3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
003360 042703 177770    ADD      #60,R3          ;CONVERT DIGIT TO ASCII.
003364 062703 000060    MOV      R3,-(1)        ;STORE ASCII CHARACTER.
003370 110341          BIC      #7,R0
003372 042700 000007    ROR      R0
003376 006000          ROR      R0
003400 006000          ROR      R0
003402 006000          DEC      R2              ;DONE ALL DIGITS?
003404 005302          BNE      OACNVA         ;BRANCH IF NOT DONE.
003406 001363          RSTO5S      ;RESTORE REGS.
003410 104007          RTI              ;DONE. EXIT.
003412 000002          ;EMT SUB TO MOVE VARIABLE NUMBER OF BYTES.
BMOVV: SAVO5S          ;SAVE REGS.
003414 104006          MOV      (5)+,R1        ;GET 'FROM' ADDRESS
003416 012501          MOV      (5)+,R2        ;GET 'TO' ADDRESS
003420 012502          MOV      (5)+,R3        ;GET COUNT
003422 012503          BMOVA: MOV      (1)+,(2)+ ;MOVE BYTE
003424 112122          DEC      R3              ;DECREMENT COUNT
003426 005303          BNE      BMOVA         ;BRANCH IF NOT DONE.
003430 001375          RSTO5S      ;RESTORE REGS.
003432 104007          RTI              ;DONE. EXIT.
003434 000002          ;EMT SUB TO CONVERT BINARY TO DECIMAL ASCII.
BDCNVV: SAVO5S          ;SAVE REGS.
003436 104006          MOV      #DECVAL,R0     ;SET UP ADDR TO STORE DECIMAL ASCII IN R0
003440 012700 003612    MOV      @((5)+,R1)      ;BINARY VALUE TO R1.
003444 013501          MOV      (5)+,BDCNVC    ;DESTINATION ADDR TO BDCNVC.
003446 012537 003522    MOV      (5)+,BDCNVD    ;CHAR COUNT TO BDCNVD.
003452 012537 003524    MOV      #ADTENP,R2     ;ADDR OF TEN POWER STRING TO R2.
003456 012702 003600    MOV      #5,CNVCTR      ;SET UP FOR 5 POWER CONVERSIONS.
003462 012737 000005 003572 BDCNVA: MOV      (2)+,TENPWR ;MOVE POWER OF TEN VALUE TO TENPWR.
003470 012237 003576    JSR      R7,SUBTEN      ;PERFORM CONVERSION
003474 004737 003532    DEC      CNVCTR          ;DONE 5 CONVERSIONS?
003500 005337 003572    BNE      BDCNVA         ;BRANCH IF NOT YET 5.
003504 001371          SUB      BDCNVD,R0
003506 163700 003524    MOV      R0,BDCNVB
003512 010037 003520    BMOVE
003516 104023          BDCNVB: OPEN
003520 000000          BDCNVC: OPEN
003522 000000          BDCNVD: OPEN
003524 000000          RSTO5S      ;RESTORE REGS.
003526 104007          RTI              ;YES. EXIT.
003530 000002          SUBTEN: CLR      DIGIT   ;CLEAR DIGIT
003532 005037 003574    SUBTNA: SUB      TENPWR,R1 ;SUBTRACT TEN POWER FROM BINARY VALUE.
003536 163701 003576    BCS      SUBTNB         ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
003542 103403          INC      DIGIT
003544 005237 003574    BR
003550 000772          SUBTNB: ADD      TENPWR,R1 ;RESTORE SUBTRACTED VALUE.
003552 063701 003576    ADD      #60,DIGIT      ;CONVERT (DIGIT) TO ASCII
003556 062737 000060 003574 MOV      DIGIT,(0)+     ;MOVE ASCII CHAR TO DECVAL FIELD.
003564 113720 003574    RTS      R7              ;EXIT.
003570 000207          CNVCTR: OPEN
003572 000000          DIGIT: OPEN
003574 000000          TENPWR: OPEN
003576 000000          ADTENP: 10000.
003600 023420

```

```

003602 001750          1000.
003604 000144          100.
003606 000012          10.
003610 000001          1
003612    040    040    040  DECVAL: .BYTE 040,040,040,040,040,040
003615    040    040    040
;EMT RANDOM NUMBER GENERATOR. NUMBER IS STORED AT LOC AFTER SUB CALL.
003620 104006          RNGEN: SAVO5S
003622 013700 003672      MOV    RP1,R0
003626 006100          ROL    R0
003630 006100          ROL    R0
003632 063700 003674      ADD    RP2,R0
003636 010037 003672      MOV    R0,RP1
003642 006100          ROL    R0
003644 006100          ROL    R0
003646 063700 003674      ADD    RP2,R0
003652 006100          ROL    R0
003654 006100          ROL    R0
003656 010037 003674      MOV    R0,RP2
003662 013725 003672      MOV    RP1,(5)+ ;STORE # AT LOC AFTER SUB CALL.
003666 104007          RSTO5S
003670 000002          RTI    ;EXIT.
003672 001233          RP1: 1233
003674 007622          RP2: 7622
;EMT SUB TO INITIALIZE RANDOM NUMBER GENERATOR.
003676 012737 001233 003672 INRNDN: MOV    #1233,RP1
003704 012737 007622 003674      MOV    #7622,RP2
003712 000002          RTI    ;EXIT.
;ROUTINE TO FETCH A CHARACTER
003714 105277 175260      AREAD: INCB   @PRS ;READER ENABLE.
003720 104031          DELAY  ;WAIT 12 MSECS FOR READER DONE.
003722 000014          12.
003724 105777 175250      TSTB   @PRS ;DONE SET?
003730 100411          BMI   ARDB ;BR IF YES.
003732 104031          DELAY  ;WAIT ADDT'L 18 MSECS.
003734 000022          18.
003736 105777 175236      TSTB   @PRS ;READER DONE NOW?
003742 100404          BMI   ARDB ;BR IF YES.
003744 104016          ERRORN ;ERROR. NO READER RESPONSE.
003746 015377          EM7
003750 177777          -1
003752 000760          BR    AREAD ;TRY AGAIN.
003754 005777 175220      ARDB:  TST   @PRS ;ERROR BIT SET?
003760 100401          BMI   ARDC ;BR IF YES.
003762 000207          RTS   %7 ;NO. EXIT.
003764 004737 003772      ARDC:  JSR   %7,TSM2 ;TYPE READER ERROR MESSAGE.
003770 000751          BR    AREAD ;TRY AGAIN.
003772 104016          TSM2: ERRORN ;TYPE READER ERROR MESSAGE.
003774 014635          SM1
003776 013243          IM6
004000 177777          -1
004002 000207          RTS   %7 ;EXIT
;ROUTINE TO SET READER INTERRUPT VECTOR AND PRIORITY
004004 017637 000000 004024 STPTRV: MOV    @(6),STPRA+2 ;MOVE VECTOR ADDR TO STPRA+2
004012 062716 000002          ADD    #2,@%6 ;SET UP EXIT
004016 013701 001210          MOV    RDRVTR,%1

```

```

004022 012721 000000 STPRA: MOV #OPEN,(1)+ ;SET VECTOR ADDRESS
004026 013721 001212 MOV RDRLVL,(1)+ ;SET PRIORITY
004032 000002 RTI ;EXIT
;ROUTINE TO SET PUNCH INTERRUPT VECTOR AND PRIORITY.
004034 017637 000000 004054 STPTPV: MOV @ (6),STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
004042 062716 000002 ADD #2,@%6 ;SET UP EXIT
004046 013701 001214 MOV PCHVTR,%1
004052 012721 000000 STPPA: MOV #OPEN,(1)+ ;SET VECTOR ADDRESS.
004056 013721 001216 MOV PCHLVL,(1)+ ;SET PRIORITY
004062 000002 RTI ;EXIT.
;SUBROUTINE TO READ CHARACTER FROM READER USING INTERRUPT.
004064 104013 BREAD: STRDRV ;SET READER VECTOR
004066 004146 BREADB ;TO BREADB
004070 052777 000101 175102 BIS #101,@PRS ;ENABLE PTR AND PTRI.
004076 005037 177776 CLR PSW ;SET UP PRTY0.
004102 012737 072460 004210 MOV #30000.,BRCTR ;DELAY APPROX. 150 MSECS.
004110 005337 004210 DEC BRCTR
004114 001375 BNE .-4
004116 005077 175056 CLR @PRS ;CLEAR PTRI ENABLE.
004122 104016 ERRORN ;TYPE NO PTR RESPONSE
004124 015377 EM7 ;MESSAGE
004126 177777 -1
004130 000755 BR BREAD ;TRY AGAIN.
004132 005077 175042 BREADA: CLR @PRS ;CLEAR READER CSR.
004136 017737 175040 001334 MOV @PRB,CRBUF ;CHAR READ TO CRBUF.
004144 000207 RTS %7 ;EXIT SUBROUTINE.
004146 022626 BREADB: POPSP2 ;RESTORE STACK.
004150 012737 000340 177776 MOV #PRTY7,PSW ;SET UP PRTY7.
004156 005777 175016 TST @PRS ;TEST FOR ERROR.
004162 100003 BPL BREADC ;BRANCH IF NO ERROR.
004164 004737 003772 JSR PC,TSM2
004170 000735 BR BREAD
004172 105777 175002 BREADC: TSTB @PRS ;TEST FOR DONE BIT.
004176 100755 BMI BREADA ;BRANCH IF DONE BIT SET.
004200 104016 ERRORN ;ERROR.FALSE READER INTERRUPT.
004202 015421 EM10
004204 177777 -1
004206 000726 BR BREAD
004210 000000 BRCTR: OPEN
;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
;DETERMINED BY CONTENTS OF LOC STLMSK.
004212 104021 STAL: RNDNUM ;GET RANDOM NUMBER.
004214 000000 STLA: OPEN ;NUMBER GOES HERE.
004216 043737 004242 004214 BIC STLMSK,STLA ;APPLY STALL MASK.
004224 001405 BEQ STALB ;BRANCH IF RESULT IS 0.
004226 013737 004214 004236 MOV STLA,STALA
004234 104031 DELAY ;DELAY
004236 000000 STALA: OPEN ;DELAY COUNT
004240 000002 STALB: RTI ;DONE. EXIT.
004242 000000 STLMSK: OPEN ;STALL MASK.
;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT (1-77)
004244 104021 GRCNT: RNDNUM ;GET RANDOM NUMBER.
004246 000000 GRCNTA: OPEN ;NUMBER GOES HERE.
004250 043737 004270 004246 BIC RCMASK,GRCNTA ;APPLY MASK.
004256 001772 BEQ GRCNT ;TRY AGAIN IF RESULT 0
004260 013737 004246 004272 MOV GRCNTA,RNCNT ;COUNT TO RNCNT

```

```

004266 000207          RTS      %7          ;EXIT.
004270 000000          RCMSK: OPEN      ;RANDOM CHARACTER MASK.
004272 000000          RNCNT: OPEN      ;RANDOM CHARACTER COUNT.
                                ;SUB TO COMPARE DATA READ FROM READER AGAINST EXPECTED DATA AND REPORT ERRORS.
004274 104027          BCHECK: GETBNR    ;GET BIN CHARACTER
004276 023737 003212 001334  CMP      BINR,CRBUF ;COMPARE BINR TO DATA IN CRBUF
004304 001001          BNE      .+4      ;BRANCH IF NOT SAME(ERROR).
004306 000207          RTS      %7          ;OK.EXIT.
004310 104017          OACNV
004312 003212          BINR
004314 015126          ASB
004316 000004          4
004320 104017          OACNV
004322 001334          CRBUF
004324 015141          AWAS
004326 000004          4
004330 104016          ERRORN
004332 015103          EM1
004334 177777          -1
004336 005337 001344          DEC      ERCTR      ;DECREMENT ERROR COUNTER
004342 001002          BNE      .+6      ;BRANCH IF NO THIRD ERROR
004344 004737 004352          JSR      %7,BSYNC  ;RESYNC THE READER.
004350 000207          RTS      %7          ;EXIT.
                                ;SUBROUTINE TO SYNC THE READER TO A SPECIAL BINARY COUNT PATTERN TEST TAPE.
004352 104026          BSYNC: INBIN      ;INITIALIZE BINARY PATTERN
004354 004737 004064          JSR      %7,BREAD  ;READ CHAR.
004360 004737 004064          JSR      %7,BREAD  ;READ CHAR.
004364 004737 004064          JSR      %7,BREAD  ;READ CHAR AND STORE AT CHR1
004370 013737 001334 001336  MOV      CRBUF,CHR1
004376 004737 004064 001336  JSR      %7,BREAD  ;READ CHAR AND STORE AT CHR2
004402 013737 001334 001340  MOV      CRBUF,CHR2
004410 004737 004064 001340  JSR      %7,BREAD  ;READ CHAR AND STORE AT CHR3.
004414 013737 001334 001342  MOV      CRBUF,CHR3
004422 004737 004440          JSR      %7,SYNCA   ;GO SYNC
004426 000751          BR      BSYNC      ;NO SYNC. TRY AGAIN.
004430 012737 000003 001344  MOV      #3,ERCTR
004436 000207          RTS      %7          ;SUCCESS.EXIT.
004440 104004          SYNCA: SAV05
004442 012700 001000          MOV      #512.,R0  ;SET UP FOR 512 TRIES.
004446 104027          SYNCB: GETBNR    ;GET BIN CHAR.
004450 013701 003212          MOV      BINR,R1   ;STORE AT R1.
004454 104027          GETBNR          ;GET BIN CHAR.
004456 013702 003212          MOV      BINR,R2   ;STORE AT R2.
004462 104027          GETBNR          ;GET BIN CHAR.
004464 013703 003212          MOV      BINR,R3   ;STORE AT R3.
004470 020137 001336          CMP      R1,CHR1   ;MATCH ON 1ST WORD?
004474 001012          BNE      SYNCC     ;BR IF NOT.
004476 020237 001340          CMP      R2,CHR2   ;MATCH ON 2ND WORD?
004502 001007          BNE      SYNCC     ;BR IF NOT.
004504 020337 001342          CMP      R3,CHR3   ;MATCH ON 3RD WORD?
004510 001004          BNE      SYNCC     ;BR IF NOT.
004512 104005          RST05
004514 062716 000002          ADD      #2,(6)    ;SET UP SINCE EXIT.
004520 000207          RTS      PC        ;EXIT.
004522 005300          SYNCC: DEC      R0  ;TRIED 512 TIMES?
004524 001350          BNE      SYNCB    ;BR IF NOT.

```

```

004526 104016          ERRORN          ;YES. SYNC ERROR.
004530 015220          EM3
004532 177777          -1
004534 104005          RST05
004536 000207          RTS          PC          ;SYNC ERROR EXIT.
;SUBROUTINE TO CHECK FOR PUNCH READY.
CPRDY: TST          @PPS          ;TEST FOR ERROR BIT.
        BMI          CPRDYA        ;BRANCH IF ERROR BIT SET.
        TSTB         @PPS          ;TEST FOR READY BIT.
        BPL          CPRDYA        ;BRANCH IF READY NOT SET.
        RTS          %7           ;OK. EXIT.
CPRDYA: TYPES          ;TYPE NOT READY MESSAGE.
        SM3
        IM16
        -1
        CHALT
        BR          CPRDY
;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: TYPES          ;TYPE SELECT PUNCH MESSAGE.
        SPCHM
        IM23
        -1
        CHALT
        CMPB         SR,PCHLIM      ;WAIT FOR USER.
        BLO          PCHSLA        ;VALID PUNCH NUMBER?
        TYPE         ;BR IF YES.
        INVRP        ;NO. TYPE MESSAGE, AND
        BR          PCHSEL         ;GO TRY AGAIN.
PCHSLA: MOV          SR,PPS          ;DEVELOP PCH CSR ADDR.
        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.
PCHSLB: OACNV        ;CONVERT SELECTED PCH NUMBER TO ASCII.
        SR
        APCHID
  
```

```

005010 000002          2
005012 104010          TYPE                ;TYPE PCH SELECTED MESSAGE.
005014 015663          PCHIDM
005016 000207          RTS          PC          ;EXIT.
;SUBROUTINE TO SELECT READER TO BE TESTED/USED.
RDRSEL: TYPES          ;TYPE SELECT READER MESSAGE.
SRDRM
IM23
-1
005020 104011          CHALT          ;WAIT FOR USER.
005022 015500          CMPB          SR,RDRLIM      ;VALID READER?
005024 014041          BLO          RDRSLA ;BR IF YES.
005026 177777          TYPE          ;NO. TYPE MESSAGE AND TRY AGAIN.
005030 104024          INVRP
005032 123737 177570 001246 BR          RDRSEL
005040 103403          MOV          SR,PRS      ;DEVELOP RDR CSR ADDR.
005042 104010          ASL          PRS
005044 015616          ASL          PRS
005046 000764          ADD          FSTRDR,PRS
005050 013737 177570 001200 RDRSLA: MOV      PRS,PRB      ;DEVELOP READER BUFFER ADDR.
005056 006337 001200          ADD          #2,PRB
005062 006337 001200          MOV          SR,RDRVTR      ;DEVELOP RDR VECTOR ADDR.
005066 063737 001220 001200 ASL          RDRVTR
005074 013737 001200 001202 ASL          RDRVTR
005102 062737 000002 001202 ADD          FSTVCT,RDRVTR
005110 013737 177570 001210 MOV          #177400,RDRMSK ;SET UP 8LEVEL READER MASK.
005116 006337 001210          TST          SR          ;8 LEVEL READER?
005122 006337 001210          BPL          RDRSLB      ;BR IF YES.
005126 063737 001244 001210 MOV          #177700,RDRMSK ;NO. SET UP 6 LEVEL MASK.
005134 012737 177400 001350 RDRSLB: OACNV      ;CONVERT SELECTED RDR NUMBER TO ASCII.
005142 005737 177570          SR
005146 100003          ARDRID
005150 012737 177700 001350 2
005156 104017          TYPE          ;TYPE RDR SELECTED MESSAGE.
005160 177570          RDRIDM
005162 015646          RTS          PC          ;EXIT.
005164 000002          CLR          ICNT      ;CLEAR ITERATION COUNT.
005166 104010          CLR          RTNNO      ;CLEAR CURRENT ROUTINE NUMBER.
005170 015641          MOV          #3,ERCTR      ;SET ERROR COUNT TO 3.
005172 000207          MOV          #300,R1      ;CLEAR INTERRUPT VECTORS.
005174 005037 001266          MOV          #302,R2
005200 005037 001262          CLNUP: CLNUPA: MOV      R2,(1)+
005204 012737 000003 001344          CLR          (1)+
005212 012701 000300          CMP          R2,1000
005216 012702 000302          BEQ          CLNUPB
005222 010221          ADD          #4,R2
005224 005021          BR          CLNUPA
005226 020237 001000          CLNUPB: RTS      PC
005232 001403          ;EXIT.
005234 062702 000004
005240 000770
005242 000207

```

```

;PRGO - READER TESTS
177777 X=-1
000000 Y=0
005244 012737 005266 001256 PRGO: MOV #POTO,KSTART ;ADDR OF 1ST ROUTINE TO KSTART.
005252 104010 TYPE ;TYPE TITLE.
005254 013267 TITLO
005256 004737 005020 JSR PC,RDRSEL ;SELECT READER.
005262 000137 001700 JMP SRSET ;GO GET STARTED.
005266 TSTA POA,1000.
005266 TSTAA POA,1000.,\X+1,\X+2,\Y
;*****
005266 000000 POTO: 0 ; PRGO ROUTINE 0 *
005270 005316 POT1 ;ADDRESS OF NEXT ROUTINE *
005272 001750 1000. ;TEST ITERATION COUNT *
005274 005304 POAA ;SCOPE ENTRY POINT *
000000 X=X+1
;*****
;TEST ABILITY TO REFERENCE THE READER STATUS WORD
005276 012737 005312 000004 POAA: MOV #POAE,MACHER ;SET UP MACHINE ERROR TRAP.
005304 005777 173670 TST @PRS ;REFERENCE READER STATUS WORD.
005310 104000 SCOPE ;SCOPE
005312 104015 POAE: ERROR ;ERROR. TRAPPED WHEN REFERENCING READER
005314 104000 SCOPE ;STATUS WORD (PRS).
005316 TSTA POB,1000.
005316 TSTAA POB,1000.,\X+1,\X+2,\Y
;*****
005316 000001 POT1: 1 ; PRGO ROUTINE 1 *
005320 005346 POT2 ;ADDRESS OF NEXT ROUTINE *
005322 001750 1000. ;TEST ITERATION COUNT *
005324 005334 POBA ;SCOPE ENTRY POINT *
000001 X=X+1
;*****
;TEST ABILITY TO REFERENCE THE READER BUFFER.
005326 012737 005342 000004 POBA: MOV #POBB,MACHER ;SET UP MACHINE ERROR TRAP.
005334 005777 173642 TST @PRB ;REFERENCE READER BUFFER
005340 104000 SCOPE ;SCOPE
005342 104015 POBB: ERROR ;ERROR. TRAPPED WHEN REFERENCING
005344 104000 SCOPE ;READER BUFFER. (PRB)
005346 TSTA POD,1000.
005346 TSTAA POD,1000.,\X+1,\X+2,\Y
;*****
005346 000002 POT2: 2 ; PRGO ROUTINE 2 *
005350 005422 POT3 ;ADDRESS OF NEXT ROUTINE *
005352 001750 1000. ;TEST ITERATION COUNT *
005354 005356 PODA ;SCOPE ENTRY POINT *
000002 X=X+1
;*****
;TEST ABILITY TO SET AND CLEAR THE ID BIT (INTERRUPT ENABLE (BIT 6))
;IN READER STATUS WORD
005356 052777 000100 173614 PODA: BIS #BIT6,@PRS ;SET ID BIT (BIT 6) IN READER PRS
005364 032777 000100 173606 BIT #BIT6,@PRS ;CHECK ID BIT IN PRS
005372 001002 BNE PODB ;ID BIT SET?
005374 104015 ERROR ;NO. ERROR. FAILED TO SET ID BIT (BIT 6)
;IN PRS.
005376 104000 SCOPE

```

```

005400 042777 000100 173572 PODB: BIC #BIT6,@PRS ;CLEAR ID BIT IN PRS.
005406 032777 000100 173564 BIT #BIT6,@PRS ;CHECK ID BIT IN PRS
005414 001401 BEQ .+4 ;BRANCH IF BIT CLEAR.
005416 104015 ERROR ;ERROR. ID BIT IN PRS FAILED TO CLEAR.
005420 104000 SCOPE
005422 TSTA POE,100.
005422 TSTAA POE,100.,\X+1,\X+2,\Y

```

```

*****
005422 000003 POT3: 3 ; PRGO ROUTINE 3 *
005424 005456 POT4 ;ADDRESS OF NEXT ROUTINE *
005426 000144 100. ;TEST ITERATION COUNT *
005430 005432 POEA ;SCOPE ENTRY POINT *
000003 X-X+1

```

```

*****
:TEST ABILITY TO CLEAR ID BIT (BIT 6) WITH RESET INSTRUCTION
005432 052777 000100 173540 POEA: BIS #BIT6,@PRS ;SET ID BIT IN PRS
005440 104001 SRESET ;RESET
005442 032777 000100 173530 BIT #BIT6,@PRS ;TEST ID BIT
005450 001401 BEQ .+4 ;BRANCH IF ID BIT IS CLEAR.
005452 104015 ERROR ;ERROR. RESET INSTRUCTION FAILED TO
005454 104000 SCOPE ;CLEAR ID BIT IN READER PRS.
005456 TSTA POF,100.
005456 TSTAA POF,100.,\X+1,\X+2,\Y

```

```

*****
005456 000004 POT4: 4 ; PRGO ROUTINE 4 *
005460 005510 POT5 ;ADDRESS OF NEXT ROUTINE *
005462 000144 100. ;TEST ITERATION COUNT *
005464 005466 POFA ;SCOPE ENTRY POINT *
000004 X=X+1

```

```

*****
:TEST THAT DONE BIT (BIT 7 OF PRS) IS SET NO LATER THAN 100 MSECS AFTER RDR ENB.
005466 005277 173506 POFA: INC @PRS ;ENABLE READER
005472 104031 DELAY ;WAIT APPROX 100 MILLISECS.
005474 000144 100.
005476 105777 173476 TSTB @PRS ;TEST FOR DONE (BIT 7)
005502 100401 BMI POFB ;BRANCH IF DONE BIT WAS SET..
005504 104015 ERROR ;ERROR, 100 MSECS AFTER READER
;ENABLE, DONE BIT WAS NOT SET.
005506 104000 POFB: SCOPE ;SCOPE
005510 TSTA POG,1000.
005510 TSTAA POG,1000.,\X+1,\X+2,\Y

```

```

*****
005510 000005 POT5: 5 ; PRGO ROUTINE 5 *
005512 005542 POT6 ;ADDRESS OF NEXT ROUTINE *
005514 001750 1000. ;TEST ITERATION COUNT *
005516 005520 POGA ;SCOPE ENTRY POINT *
000005 X=X+1

```

```

*****
:TEST ABILITY TO READ DONE BIT (BIT 7 OF PRS) RELIABLY
005520 005277 173454 POGA: INC @PRS ;ENABLE READER
005524 104031 DELAY ;WAIT APPROX 100 MILLISECS
005526 000144 100.
005530 105777 173444 TSTB @PRS ;TEST DONE BIT (BIT 7 OF PRS)
005534 100401 BMI .+4 ;BRANCH IF DONE BIT SET
005536 104015 ERROR ;ERROR. DONE BIT NOT SET, OR FAILED
005540 104000 SCOPE ;TO READ IT.

```



```
005542          TSTA   POH,100.
005542          TSTAA  POH,100.,\X+1,\X+2,\Y
:*****
005542 000006   POT6:  6          ; PRGO ROUTINE 6          *
005544 005612   POT7          ; ADDRESS OF NEXT ROUTINE *
005546 000144   100.          ; TEST ITERATION COUNT   *
005550 005552   POHA          ; SCOPE ENTRY POINT      *
005550 000006   X=X+1
:*****
:TEST THAT RESET COMMAND CLEARS DONE BIT (BIT 7 OF PRS)
005552 005277 173422 POHA:  INC      @PRS          ; ENABLE READER
005556 104031   DELAY          ; DELAY APPROX 50 MILLISECONDS
005560 000062   50.
005562 105777 173412 TSTB   @PRS          ; TEST FOR DONE BIT
005566 100005   BPL   POHB     ; BRANCH IF DONE BIT NOT SET
005570 000005   RESET        ; RESET
005572 105777 173402 TSTB   @PRS          ; TEST DONE BIT
005576 100403   BMI   POHC     ; BRANCH IF DONE BIT STILL SET.
005600 104000   SCOPE        ; SCOPE
005602 104015   POHB:  ERROR   ; ERROR 1. DONE BIT NOT SET.
005604 104000   SCOPE
005606 104015   POHC:  ERROR   ; ERROR 2. DONE BIT NOT RESET BY
005610 104000   SCOPE        ; RESET INSTRUCTION.
005612          TSTA   POI,100.
005612          TSTAA  POI,100.,\X+1,\X+2,\Y
:*****
005612 000007   POT7:  7          ; PRGO ROUTINE 7          *
005614 005654   POT10         ; ADDRESS OF NEXT ROUTINE *
005616 000144   100.          ; TEST ITERATION COUNT   *
005620 005622   POIA          ; SCOPE ENTRY POINT      *
005620 000007   X=X+1
:*****
:TEST THAT DONE BIT (BIT 7 OF PRS) IS CLEARED WHEN ENABLING THE READER
005622 104001   POIA:  SRESET  ; RESET
005624 005277 173350   INC      @PRS          ; ENABLE READER
005630 105777 173344   TSTB   @PRS          ; TEST FOR DONE BIT
005634 100375   BPL   .-4      ; BRANCH IF DONE BIT NOT SET
005636 005277 173336   INC      @PRS          ; ENABLE READER AGAIN
005642 105777 173332   TSTB   @PRS          ; TEST DONE BIT AGAIN
005646 100001   BPL   .+4      ; BRANCH IF DONE BIT IS RESET
005650 104015   ERROR        ; READER ENABLE DID NOT CLEAR DONE BIT
005652 104000   SCOPE        ; SCOPE
005654          TSTA   POJ,100.
005654          TSTAA  POJ,100.,\X+1,\X+2,\Y
:*****
005654 000010   POT10: 10         ; PRGO ROUTINE 10        *
005656 005714   POT11         ; ADDRESS OF NEXT ROUTINE *
005660 000144   100.          ; TEST ITERATION COUNT   *
005662 005664   POJA          ; SCOPE ENTRY POINT      *
005662 000010   X=X+1
:*****
:TEST THAT DONE BIT IS CLEARED BY REFERENCING READER BUFFER (PRB)
005664 005277 173310 POJA:  INC      @PRS          ; ENABLE READER
005670 105777 173304   TSTB   @PRS          ; TEST FOR DONE BIT
005674 100375   BPL   .-4      ; BRANCH IF DONE BIT NOT SET.
005676 005777 173300   TST    @PRB         ; REFERENCE READER BUFFER (PRB)
```

```

005702 105777 173272      TSTB @PRS      ;TEST FOR DONE BIT
005706 100001      BPL .+4       ;BRANCH IF DONE BIT IS CLEAR.
005710 104015      ERROR        ;ERROR 1. DONE BIT WAS NOT CLEARED
005712 104000      SCOPE        ;BY REFERENCING READER BUFFER.
005714      TSTA POM,100.
005714      TSTAA POM,100.,\X+1,\X+2,\Y
;*****
005714 000011      POT11: 11      ; PRGO ROUTINE 11 *
005716 006016      POT12      ;ADDRESS OF NEXT ROUTINE *
005720 000144      100.        ;TEST ITERATION COUNT *
005722 005724      POMA      ;SCOPE ENTRY POINT *
005722 000011      X=X+1
;*****
;TEST ABILITY TO READ READER BUFFER RELIABLY.
005724 012700 000144      POMA: MOV #100.,%0    ;SET COUNT TO 100 IN R0
005730 004737 003714      JSR %7,AREAD      ;GET CHARACTER
005734 017737 173242 001336      MOV @PRB,CHR1     ;C(PRB) TO CHR1
005742 017737 173234 001340      POMB: MOV @PRB,CHR2  ;C(PRB) TO CHR2
005750 023737 001336 001340      CMP CHR1,CHR2    ;COMPARE CHR1 AND CHR2.
005756 001003      BNE POMC      ;BRANCH IF R1 AND R2 DON'T MATCH
005760 005300      DEC %0
005762 001367      BNE POMB
005764 104000      SCOPE      ;SCOPE
005766 104017      POMC: OACNV     ;CORRECT 1ST READ DATA TO ASCII
005770 001336      CHR1
005772 015200      ORGRD
005774 000004      4
005776 104017      OACNV
006000 001340      CHR2
006002 015213      SUBRD
006004 000004      4
006006 104016      ERRORN     ;ERROR. REREAD OF PRB DID NOT MATCH
006010 015146      EM2        ;INITIAL DATA READ FROM PRB.
006012 177777      -1
006014 104000      SCOPE
006016      TSTA PON,100.
006016      TSTAA PON,100.,\X+1,\X+2,\Y
;*****
006016 000012      POT12: 12      ; PRGO ROUTINE 12 *
006020 006060      POT13      ;ADDRESS OF NEXT ROUTINE *
006022 000144      100.        ;TEST ITERATION COUNT *
006024 006026      PONA      ;SCOPE ENTRY POINT *
006024 000012      X=X+1
;*****
;TEST THAT READER BUFFER (PRB) IS CLEARED BY READER ENABLE
006026 104001      PONA: SRESET    ;RESET
006030 004737 003714      JSR %7,AREAD      ;GET CHARACTER
006034 005777 173142      TST @PRB        ;TEST CONTENTS OF READER BUFFER.
006040 001772      BEQ PONA      ;GO GET ANOTHER CHAR IF 0.
006042 005277 173132      INC @PRS        ;NOT 0. ENABLE READER
006046 005777 173130      TST @PRB        ;CHECK PRB
006052 001401      BEQ .+4       ;BRANCH IF PRB IS RESET
006054 104015      ERROR      ;ERROR. PRB NOT RESET BY READER ENABLE.
006056 104000      SCOPE
006060      TSTA P00,100.
006060      TSTAA PC0,100.,\X+1,\X.2,\Y

```

```

:*****
006060 000013 POT13: 13 ; PRGO ROUTINE 13 *
006062 006130 POT14 ; ADDRESS OF NEXT ROUTINE *
006064 000144 100. ; TEST ITERATION COUNT *
006066 006074 POQA ; SCOPE ENTRY POINT *
000013 X=X+1
:*****
;TEST THAT READER IS ABLE TO INTERRUPT. IF INTERRUPT IS SERVICED, IT WILL
;HAVE OCCURRED AT CORRECT VECTOR.
006070 104013 STRDRV ;SET UP READER INTERRUPT VECTOR
006072 006126 POQB
006074 012737 000000 177776 POQA: MOV #PRTY0,PSW ;SET PROCESSOR PRIORITY TO 0
006102 042777 000100 173070 BIC #BIT6,@PRS ;DISABLE READER INTERRUPT.
006110 004737 003714 JSR %7,AREAD ;GO READ CHARACTER.
006114 052777 000100 173056 BIS #BIT6,@PRS ;ENABLE READER INTERRUPT.
006122 000240 NOP ;NO OP
006124 104015 ERROR ;ERROR. READER FAILED TO INTERRUPT.
006126 104000 POQB: SCOPE ;SCOPE
006130 TSTA POP,100.
006130 TSTAA POP,100.,\X+1,\X+2,\Y
:*****
006130 000014 POT14: 14 ; PRGO ROUTINE 14 *
006132 006204 POT15 ; ADDRESS OF NEXT ROUTINE *
006134 000144 100. ; TEST ITERATION COUNT *
006136 006144 POPA ; SCOPE ENTRY POINT *
000014 X=X+1
:*****
;TEST THAT READER DOES NOT INTERRUPT WITH PROCESSOR AT SAME PRIORITY
;LEVEL AS READER.
006140 104013 STRDRV ;SET UP READER INTERRUPT VECTOR
006142 006200 POPE
006144 013737 001212 177776 POPA: MOV RDRLVL,PSW ;SET PROCESSOR PRIORITY SAME AS READER PRIORITY.
006152 005077 173022 CLR @PRS ;DISABLE READER INTERRUPT.
006156 004737 003714 JSR %7,AREAD ;GO READ A CHARACTER.
006162 052777 000100 173010 BIS #BIT6,@PRS ;ENABLE READER INTERRUPT.
006170 000240 NOP ;OK IF NO INTERRUPT OCCURS.
006172 005077 173002 CLR @PRS ;DISABLE READER INTERRUPT.
006176 104000 SCOPE ;SCOPE
006200 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 RDRLVL.
006202 104000 SCOPE
006204 TSTA POP,100.
006204 TSTAA POP,100.,\X+1,\X+2,\Y
:*****
006204 000015 POT15: 15 ; PRGO ROUTINE 15 *
006206 006262 POT16 ; ADDRESS OF NEXT ROUTINE *
006210 000144 100. ; TEST ITERATION COUNT *
006212 006220 POQA ; SCOPE ENTRY POINT *
000015 X=X+1
:*****
;TEST THAT READER INTERRUPTS WITH PROCESSOR AT PRIORITY 1 LEVEL LOWER
;THAN READER'S
006214 104013 STRDRV ;SET UP READER INTERRUPT VECTOR
006216 006260 POQB
006220 013737 001212 177776 POQA: MOV RDRLVL,PSW ;SET PROCESSOR PRIORITY ONE LEVEL LOWER

```

```

006226 162737 000040 177776 SUB #40,PSW ;THAN READER PRIORITY
006234 042777 000100 172736 BIC #BIT6,@PRS ;DISABLE READER INTERRUPT
006242 004737 003714 JSR %7,AREAD ;GO READ A CHARACTER.
006246 052777 000100 172724 BIS #BIT6,@PRS ;ENABLE READER INTERRUPT
006254 000240 NOP ;NOP
006256 104015 ERROR ;READER FAILED TO INTERRUPT WITH
;PROCESSOR PRIORITY ONE LEVEL LOWER THAN
;READER. THEREFORE, READER PRIORITY MUST BE
;LOWER THAN SPECIFIED AT RDRLVL

```

```

006260 104000 POQB: SCOPE
006262 TSTA POR,100.
006262 TSTAA POR,100.,\X+1,\X+2,\Y

```

```

;*****
006262 000016 POT16: 16 ; PRGO ROUTINE 16 *
006264 006354 POT17 ; ADDRESS OF NEXT ROUTINE *
006266 000144 100. ; TEST ITERATION COUNT *
006270 006272 PORA ; SCOPE ENTRY POINT *
000016 X=X+1

```

```

;*****
;TEST THAT READER DOES NOT REINTERRUPT AFTER RTI WHEN DONE BIT IS NOT CLEARED

```

```

006272 104013 PORA: STRDRV ;SET READER INTERRUPT VECTOR
006274 006330 PORC
006276 012737 000000 177776 MOV #PRTY0,PSW ;SET PROCESSOR TO PRIORITY 0
006304 005077 172670 CLR @PRS ;DISABLE READER INTERRUPT.
006310 004737 003714 JSR %7,AREAD ;GO READ A CHARACTER.
006314 052777 000100 172656 BIS #BIT6,@PRS ;ENABLE READER INTERRUPT

```

```

006322 000240 NOP
006324 104015 ERROR ;ERROR 1. READER FAILED TO INTERRUPT
006326 104000 SCOPE ;SCOPE
006330 012777 006350 172652 PORC: MOV #PORE,@RDRVTR ;CHANGE INTERRUPT VECTOR TO PORE
006336 012716 006344 MOV #PORD,%6
006342 000002 RTI ;RETURN FROM INTERRUPT

```

```

006344 000240 PORD: NOP
006346 104000 SCOPE
006350 104015 PORE: ERROR ;ERROR 2. READER REINTERRUPTED AFTER
006352 104000 SCOPE ;RTI WITH DONE BIT LEFT ON
006354 TSTA POS,1000.
006354 TSTAA POS,1000.,\X+1,\X+2,\Y

```

```

;*****
006354 000017 POT17: 17 ; PRGO ROUTINE 17 *
006356 006424 POT20 ; ADDRESS OF NEXT ROUTINE *
006360 001750 1000. ; TEST ITERATION COUNT *
006362 006370 POSA ; SCOPE ENTRY POINT *
000017 X=X+1

```

```

;*****
;TEST THAT READER INTERRUPTS IMMEDIATELY UPON LOWERING CP PRIORITY TO 0.

```

```

006364 104013 STRDRV ;SET READER INTERRUPT VECTOR TO
006366 006422 POSB
006370 005077 172604 POSA: CLR @PRS ;DISABLE PTRI.
006374 004737 003714 JSR %7,AREAD ;READ A CHARACTER.
006400 052777 000100 172572 BIS #BIT6,@PRS ;ENABLE PTRI
006406 005037 177776 CLR PSW ;LOWER PRIORITY TO 0.
006412 012737 000340 177776 MOV #PRTY7,PSW ;RAISE PRIORITY BACK TO 7.
006420 104015 ERROR ;ERROR. READER FAILED TO INTERRUPT IMMEDIATELY
;AFTER LOWERING PRIORITY TO 0

```

```

006422 104000 POSB: SCOPE ;INTERRUPTS TO HERE IF SUCCESSFUL.
006424 TSTA POT,10000.

```

```
006424          TSTAA  POT,10000.,\X+1,\X+2,\Y
:*****
006424 000020    POT20: 20          ; PRGO ROUTINE 20          *
006426 006452    POT21          ; ADDRESS OF NEXT ROUTINE *
006430 023420    10000.        ; TEST ITERATION COUNT   *
006432 006440    POTA          ; SCOPE ENTRY POINT      *
006432 000020    X=X+1
:*****
006434 004737 004352    ; READ AND CHECK 10000 CHARACTERS OF SPECIAL BINARY COUNT PATTERN. FULL SPEED.
006440 004737 004064    POTA: JSR  %7,BSYNC        ; SYNC READER; SET ERROR COUNTER.
006444 004737 004274    JSR  %7,BREAD          ; GO READ CHARACTER
006450 104000    JSR  %7,BCHECK        ; GO CHECK CHARACTER READ.
006452          SCOPE          ; SCOPE
006452          TSTA  POU,500.
006452          TSTAA POU,500.,\X+1,\X+2,\Y
:*****
006452 000021    POT21: 21          ; PRGO ROUTINE 21          *
006454 006510    POT22          ; ADDRESS OF NEXT ROUTINE *
006456 000764    500.          ; TEST ITERATION COUNT   *
006460 006474    POUA          ; SCOPE ENTRY POINT      *
006460 000021    X=X+1
:*****
006462 012737 177770 004242    ; READ AND CHECK 500 CHARACTERS OF SPECIAL BINARY COUNT PATTERN.
006470 004737 004352    ; RANDOM STALL BETWEEN CHARACTERS (0 TO 7 MSECS).
006474 104012    MOV  #177770,STLMSK
006476 004737 004064    POUA: JSR  %7,BSYNC        ; SYNC READER; SET ERROR COUNTER
006502 004737 004274    STALL          ; RANDOM STALL (0 TO 7 MSECS)
006506 104000    JSR  %7,BREAD          ; GO READ CHARACTER
006510          JSR  %7,BCHECK        ; GO CHECK CHARACTER READ
006510          SCOPE          ; SCOPE
006510          TSTA  POV,1000.
006510          TSTAA POV,1000.,\X+1,\X+2,\Y
:*****
006510 000022    POT22: 22          ; PRGO ROUTINE 22          *
006512 006562    POT23          ; ADDRESS OF NEXT ROUTINE *
006514 001750    1000.         ; TEST ITERATION COUNT   *
006516 006532    POVA          ; SCOPE ENTRY POINT      *
006516 000022    X-X+1
:*****
006520 012737 177740 004242    ; READ 1000 GROUPS OF 3 CHARS EACH. STALL (0 TO 31 MSECS) BEFORE EACH GROUP.
006526 004737 004352    MOV  #177740,STLMSK ; LIMIT STALLS TO 31 MSECS.
006532 012737 000003 004272    POVA: JSR  %7,BSYNC        ; SYNC READER. SET ERROR COUNTER
006540 104012    MOV  #3,RNCNT        ; SET CHAR COUNT TO 3.
006542 004737 004064    STALL          ; RANDOM STALL (0 TO 31 MSECS).
006546 004737 004274    POVB: JSR  %7,BREAD          ; GO READ CHARACTER.
006552 005337 004272    JSR  %7,BCHECK        ; GO CHECK CHARACTER READ.
006556 001371    DEC  RNCNT          ; 3 CHARS READ?
006560 104000    BNE  POVB          ; BR IF NOT 3 CHARS YET.
006562          SCOPE          ; SCOPE
006562          TSTA  POX,1000.
006562          TSTAA POX,1000.,\X+1,\X+2,\Y
:*****
006562 000023    POT23: 23          ; PRGO ROUTINE 23          *
006564 006640    POT24          ; ADDRESS OF NEXT ROUTINE *
006566 001750    1000.         ; TEST ITERATION COUNT   *
006570 006612    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.
006572 012737 177740 004242      MOV      #177740,STLMSK ;LIMIT STALLS TO 31 MSECS.
006600 012737 177760 004270      MOV      #177760,RCMSK ;LIMIT MAY CHAR COUNT TO 15 CHARS.
006606 004737 004352              JSR      %7,BSYNC   ;SYNC READER. SET ERROR COUNTER.
006612 004737 004244      POXA:  JSR      %7,GRCNT   ;GENERATE RANDOM CHAR COUNT.
006616 104012                    STALL
006620 004737 004064      POXB:  JSR      %7,BREAD   ;GO READ CHARACTER.
006624 004737 004274              JSR      %7,BCHECK  ;GO CHECK CHARACTER.
006630 005337 004272              DEC      RNCNT     ;ALL CHARS READ?
006634 001371                    BNE     POXB      ;BRANCH IF NOT.
006636 104000                    SCOPE
006640                    TSTAA   POY,1000.,\X+1,LST,\Y
:*****
006640 000024      POT24:  24          ; PRG0 ROUTINE 24 *
006642 177777      POTLST ;ADDRESS OF NEXT ROUTINE *
006644 001750      1000.   ;TEST ITERATION COUNT *
006646 006670      POYA    ;SCOPE ENTRY POINT *
000024          X=X+1
:*****
: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).
006650 012737 177740 004242      MOV      #177740,STLMSK
006656 012737 177700 004270      MOV      #177700,RCMSK
006664 004737 004352              JSR      %7,BSYNC   ;SYNC READER; SET ERROR COUNTER.
006670 004737 004244      POYA:  JSR      %7,GRCNT   ;GENERATE RANDOM CHARACTER COUNT.
006674 104012                    STALL ;RANDOM STALL (0 TO 31MSECS)
006676 004737 004064      POYB:  JSR      %7,BREAD   ;GO READ CHARACTER
006702 004737 004274              JSR      %7,BCHECK  ;GO CHECK CHARACTER READ
006706 005337 004272              DEC      RNCNT     ;DECREMENT RANDOM CHAR COUNT
006712 001371                    BNE     POYB      ;GO READ AGAIN IF COUNT NOT 0.
006714 104000                    SCOPE ;SCOPE
:PRG1 - PUNCH TESTS
          X=-1
          Y=1
006716 012737 006740 001256      PRG1:  MOV      #P1T0,KSTART ;ADDR OF 1ST ROUTINE TO KSTART
006724 104010                    TYPE   ;TYPE TITLE.
006726 013151                    TITL1
006730 004737 004622              JSR      PC,PCHSEL  ;SELECT PUNCH.
006734 000137 001700              JMP      SRSET     ;GO GET STARTED.
006740                    TSTA   P1A,1000.
006740                    TSTAA  P1A,1000.,\X+1,\X+2,\Y
:*****
006740 000000      P1T0:  0          ; PRG1 ROUTINE 0 *
006742 006770      P1T1   ;ADDRESS OF NEXT ROUTINE *
006744 001750      1000.   ;TEST ITERATION COUNT *
006746 006756      P1AA   ;SCOPE ENTRY POINT *
000000          X=X+1
:*****
:TEST ABILITY TO REFERENCE THE PUNCH STATUS WORD (PPS)
006750 012737 006764 000004      P1AA:  MOV      #P1AB,MACHER ;REFERENCE PUNCH STATUS WORD
006756 005777 172222              TST     @PPS
006762 104000                    SCOPE ;SCOPE

```

```
006764 104015 P1AB: ERROR ;ERROR. TRAPPED WHEN REFERENCING PUNCH
006766 104000 SCOPE ;STATUS WORD (PPS).
006770 TSTA P1B,1000.
006770 TSTAA P1B,1000.,\X+1,\X+2,\Y
;*****
006770 000001 P1T1: 1 ; PRG1 ROUTINE 1 *
006772 007020 P1T2 ;ADDRESS OF NEXT ROUTINE *
006774 001750 1000. ;TEST ITERATION COUNT *
006776 007006 P1BA ;SCOPE ENTRY POINT *
000001 X=X+1
;*****
007000 012737 007014 000004 ;TEST ABILITY TO REFERENCE THE PUNCH BUFFER (PPB)
007006 005777 172174 MOV #P1BB,MACHER ;SET UP MACHINE ERROR TRAP.
007012 104000 P1BA: TST @PPB ;REFERENCE PUNCH BUFFER.
007014 104015 SCOPE ;SCOPE
007016 104000 P1BB: ERROR ;TRAPPED WHEN REFERENCING
SCOPE ;PUNCH BUFFER (PPB)
007020 TSTA P1C,1000.
007020 TSTAA P1C,1000.,\X+1,\X+2,\Y
;*****
007020 000002 P1T2: 2 ; PRG1 ROUTINE 2 *
007022 007074 P1T3 ;ADDRESS OF NEXT ROUTINE *
007024 001750 1000. ;TEST ITERATION COUNT *
007026 007030 P1CA ;SCOPE ENTRY POINT *
000002 X=X+1
;*****
007030 052777 000100 172146 ;TEST ABILITY TO SET AND CLEAR ID BIT (BIT 6) IN PPS
007036 032777 000100 172140 P1CA: BIS #BIT6,@PPS ;SET ID BIT IN PPS (BIT 6)
BIT #BIT6,@PPS ;CHECK ID BIT IN PPS
007044 001002 BNE P1CB ;BRANCH IF BIT SET
007046 104015 ERROR ;ERROR. FAILED TO SET ID BIT (BIT 6) IN
007050 104000 SCOPE ;PPS
007052 042777 000100 172124 P1CB: BIC #BIT6,@PPS ;CLEAR ID BIT IN PPS
007060 032777 000100 172116 BIT #BIT6,@PPS ;CHECK ID BIT IN PPS
007066 001401 BEQ .+4 ;BRANCH IF BIT IS CLEAR
007070 104015 ERROR ;ERROR. ID BIT IN PPS FAILED TO CLEAR
007072 104000 SCOPE
007074 TSTA P1D,100.
007074 TSTAA P1D,100.,\X+1,\X+2,\Y
;*****
007074 000003 P1T3: 3 ; PRG1 ROUTINE 3 *
007076 007130 P1T4 ;ADDRESS OF NEXT ROUTINE *
007100 000144 100. ;TEST ITERATION COUNT *
007102 007104 P1DA ;SCOPE ENTRY POINT *
000003 X=X+1
;*****
007104 052777 000100 172072 ;TEST ABILITY TO CLEAR ID BIT (6) IN PPS
007112 104001 P1DA: BIS #BIT6,@PPS ;SET ID BIT IN PPS.
SRESET ;RESET.
007114 032777 000100 172062 BIT #BIT6,@PPS ;TEST ID BIT IN PPS.
007122 001401 BEQ .+4 ;BRANCH IF ID BIT IS CLEAR
007124 104015 ERROR ;ERROR. RESET INSTRUCTION FAILED TO
007126 104000 SCOPE ;CLEAR ID BIT (BIT 6) IN PPS.
007130 TSTA P1E,1000.
007130 TSTAA P1E,1000.,\X+1,\X+2,\Y
;*****
```

```
007130 000004          P1T4:  4          ; PRG1 ROUTINE 4          *
007132 007152          P1T5          ; ADDRESS OF NEXT ROUTINE *
007134 001750          1000.         ; TEST ITERATION COUNT   *
007136 007140          P1EA          ; 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.
007140 105777 172040  P1EA:  TSTB  @PPS          ; TEST PPS
007144 100401          BMI    .+4          ; BRANCH IF READY BIT SET
007146 104015          ERROR          ; ERROR. RESET FAILED TO SET READY BIT,
007150 104000          SCOPE          ; OR FAILED TO READ READY BIT.
007152          TSTA  P1F,100.
007152          TSTAA P1F,100.,\X+1,\X+2,\Y
;*****
007152 000005          P1T5:  5          ; PRG1 ROUTINE 5          *
007154 007210          P1T6          ; ADDRESS OF NEXT ROUTINE *
007156 000144          100.         ; TEST ITERATION COUNT   *
007160 007162          P1FA          ; SCOPE ENTRY POINT      *
          000005          X=X+1
;*****
;TEST THAT READY BIT (BIT 7) OF PPS IS RESET BY LOADING PUNCH BUFFER (PPB)
007162 104001          P1FA:  SRESET          ; RESET
007164 004737 004540  JSR    %7,CPRDY        ; CHECK FOR PUNCH READY
007170 112777 000000 212010  MOVB  #0,@PPB          ; LOAD 0 INTO PUNCH BUFFER (PPB)
007176 105777 172002  TSTB  @PPS          ; TEST PPS
007202 100001          BPL    .+4          ; BRANCH IF READY BIT CLEAR
007204 104015          ERROR          ; ERROR. LOADING PUNCH BUFFER (PPB)
007206 104000          SCOPE          ; FAILED TO RESET READY BIT IN PPS
007210          TSTA  P1G,100.
007210          TSTAA P1G,100.,\X+1,\X+2,\Y
;*****
007210 000006          P1T6:  6          ; PRG1 ROUTINE 6          *
007212 007252          P1T7          ; ADDRESS OF NEXT ROUTINE *
007214 000144          100.         ; TEST ITERATION COUNT   *
007216 007220          P1GA          ; SCOPE ENTRY POINT      *
          000006          X=X+1
;*****
;TEST THAT READY BIT (BIT 7) IS NOT RESET BY BYTE LOADING PPB+1.
007220 104001          P1GA:  SRESET          ; RESET
007222 004737 004540  JSR    %7,CPRDY        ; CHECK FOR PUNCH READY.
007226 013700 001206  MOV   PPB,%0
007232 005200          INC    %0
007234 112710 000000  MOVB  #0,@%0          ; LOAD PPB+1
007240 105777 171740  TSTB  @PPS          ; TEST PPS
007244 100401          BMI    .+4          ; BRANCH IF READY BIT NOT RESET.
007246 104015          ERROR          ; ERROR. LOADING PPB+1 CLEARED READY BIT.
007250 104000          SCOPE          ; SCOPE
007252          TSTA  P1H,1000.
007252          TSTAA P1H,1000.,\X+1,\X+2,\Y
;*****
007252 000007          P1T7:  7          ; PRG1 ROUTINE 7          *
007254 007320          P1T10         ; ADDRESS OF NEXT ROUTINE *
007256 001750          1000.         ; TEST ITERATION COUNT   *
007260 007266          P1HA          ; 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.
007262 104014 STPCHV ;SET UP PUNCH INTERRUPT VECTOR.
007264 007316 P1HB
007266 005037 177776 P1HA: CLR PSW ;SET PRY TO 0.
007272 004737 004540 JSR %7,CPRDY ;CHECK FOR PUNCH READY.
007276 042777 000100 171700 BIC #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
007304 052777 000100 171672 BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT
007312 000240 NOP
007314 104015 ERROR ;ERROR. FAILURE TO INTERRUPT WITH
;PUNCH READY BIT SET.
007316 104000 P1HB: SCOPE ;INTERRUPT VECTOR POINTS HERE.
007320 TSTA P1I,1000.
007320 TSTAA P1I,1000.,\X+1,\X+2,\Y
:*****
P1T10: 10 ; PRG1 ROUTINE 10 *
007322 007412 P1T11 ;ADDRESS OF NEXT ROUTINE *
007324 001750 1000. ;TEST ITERATION COUNT *
007326 007330 P1JA ;SCOPE ENTRY POINT *
000010 X=X+1
:*****
:TEST THAT PUNCH DOES NOT REINTERRUPT AFTER RTI WHEN READY BIT IS NOT RESET.
P1IA: STPCHV ;SET UP PUNCH INTERRUPT VECTOR
007330 104014 P1IB
007332 007366 CLR PSW ;SET PRY TO 0.
007334 005037 177776 JSR %7,CPRDY ;CHECK FOR PUNCH READY.
007340 004737 004540 BIC #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
007344 042777 000100 171632 BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT
007352 052777 000100 171624 NOP
007360 000240 ERROR ;ERROR. PUNCH FAILED TO INTERRUPT.
007362 104015 SCOPE ;SCOPE
007364 104000 P1IB: MOV #P1ID,@PCHVTR ;CHANGE INTERRUPT VECTOR TO P1ID
007366 012777 007406 171620 MOV #P1IC,@%6 ;CHANGE INTERRUPT RETURN ADDRESS.
007374 012716 007402 RTI ;RETURN FROM INTERRUPT.
007400 000002 P1IC: NOP
007402 000240 P1ID: ERROR ;ERROR2. PUNCH REINTERRUPTED AFTER RTI WITH
007404 104000 SCOPE ;READY BIT LEFT ON. POP THE STOCK TWICE
007406 104015 TSTA P1J,1000.
007410 104000 TSTAA P1J,1000.,\X+1,\X+2,\Y
007412 000011 P1T11: 11 ; PRG1 ROUTINE 11 *
007414 007466 P1T12 ;ADDRESS OF NEXT ROUTINE *
007416 001750 1000. ;TEST ITERATION COUNT *
007420 007426 P1JA ;SCOPE ENTRY POINT *
000011 X=X+1
:*****
:TEST THAT THE PUNCH DOES NOT INTERRUPT WITH PROCESSOR AT SAME PRIORITY
:LEVEL AS THE PUNCH.
007422 104014 STPCHV ;SET UP PUNCH INTERRUPT VECTOR.
007424 007462 P1JB
007426 013737 001216 177776 P1JA: MOV PCHLVL,PSW ;SET PROCESSOR PRIORITY SAME AS PUNCH.
007434 005077 171544 CLR @PPS ;DISABLE PUNCH INTERRUPT.
007440 004737 004540 JSR %7,CPRDY ;CHECK FOR PUNCH READY.
007444 052777 000100 171532 BIS #BIT6,@PPS ;ENABLE PUNCH INTERRUPT.

```

```

007452 000240      NOP      ;OK IF NO INTERRUPT OCCURS.
007454 005077 171524 CLR      @PPS  ;DISABLE PUNCH INTERRUPT.
007460 104000      SCOPE   ;SCOPE
007462 104015      P1JB:  ERROR  ;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.

007464 104000      SCOPE
007466          TSTA   P1K,1000.
007466          TSTAA  P1K,1000.,\X+1,\X+2,\Y
;*****
P1T12: 12          ; PRG1 ROUTINE 12 *
007470 007544      P1T13  ;ADDRESS OF NEXT ROUTINE *
007472 001750      1000.  ;TEST ITERATION COUNT *
007474 007502      P1KA   ;SCOPE ENTRY POINT *
          000012      X=X+1
;*****
;TEST THAT PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY 1 LEVEL LOWER
;THAN THE PUNCH PRIORITY.
007476 104014      STPCHV ;SET UP PUNCH INTERRUPT VECTOR
007500 007542      P1KB
007502 013737 001216 177776 P1KA: MOV   PCHLVL,PSW ;SET PROCESSOR PRIORITY ONE LEVEL LOWER
007510 162737 000040 177776 SUB   #40,PSW ;THAN PUNCH PRIORITY.
007516 042777 000100 171460 BIC   #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
007524 004737 004540      JSR   %7,CPRDY ;CHECK FOR PUNCH READY.
007530 052777 000100 171446 BIS   #BIT6,@PPS ;ENABLE PUNCH INTERRUPT.
007536 000240      NOP
007540 104015      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.

007542 104000      P1KB:  SCOPE
007544          TSTA   P1L,1000.
007544          TSTAA  P1L,1000.,\X+1,\X+2,\Y
;*****
P1T13: 13          ; PRG1 ROUTINE 13 *
007546 007616      P1T14  ;ADDRESS OF NEXT ROUTINE *
007550 001750      1000.  ;TEST ITERATION COUNT *
007552 007560      P1LA   ;SCOPE ENTRY POINT *
          000013      X=X+1
;*****
;TEST THAT PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING CP PRIORITY TO 0.
007554 104014      STPCHV ;SET UP PUNCH INTERRUPT VECTOR
007556 007614      P1LB
007560 004737 004540      P1LA: JSR   %7,CPRDY ;CHECK FOR PUNCH READY.
007564 042777 000100 171412 BIC   #BIT6,@PPS ;DISABLE PUNCH INTERRUPT
007572 052777 000100 171404 BIS   #BIT6,@PPS ;ENABLE PUNCH INTERRUPT
007600 005037 177776      CLR   PSW ;LOWER PRTY TO 0.
007604 012737 000340 177776 MOV   #PRTY7,PSW ;RAISE CP PRIORITY BACK TO 7.
007612 104015      ERROR  ;ERROR. PUNCH FAILED TO INTERRUPT IMMEDIATELY
                                ;AFTER CP PRIORITY WAS LOWERED TO 0.
                                ;HERE IF INTERRUPT OCCURS.

007614 104000      P1LB:  SCOPE
007616          TSTA   P1M,5
007616          TSTAA  P1M,5,\X+1,\X+2,\Y
;*****
P1T14: 14          ; PRG1 ROUTINE 14 *
007620 007662      P1T15  ;ADDRESS OF NEXT ROUTINE *

```

```
007622 000005          5          ;TEST ITERATION COUNT          *
007624 007626          P1MA          ;SCOPE ENTRY POINT          *
          000014          X=X+1
;*****
;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 0 (FULL SPEED)
007626 012737 001000 001330 P1MA:  MOV  #512.,RCNT  ;SET CHARACTER COUNT TO 512
007634 004537 010122          JSR   %5,PFRNT  ;GO PUNCH FRONT END AND MODE 0
007640 000000          0          ;INDICATOR
007642 104026          INBIN          ;INITIALIZE SPECIAL BINARY COUNT
007644 104030          P1MB:  GETBNP          ;GET BINARY CHARACTER
007646 004737 004572          JSR   %7,HSPCH  ;GO PUNCH THE CHARACTER
007652 005337 001330          DEC   RCNT      ;DECREMENT CHAR COUNT.
007656 001372          BNE   P1MB      ;BRANCH IF COUNT NOT YET 0 YET.
007660 104000          SCOPE          ;SCOPE
007662          TSTA  P1N,5
007662          TSTAA P1N,5,\X+1,\X+2,\Y
;*****
007662 000015          P1T15: 15          ; PRG1 ROUTINE 15          *
007664 007736          P1T16          ;ADDRESS OF NEXT ROUTINE          *
007666 000005          5          ;TEST ITERATION COUNT          *
007670 007700          P1NA          ;SCOPE ENTRY POINT          *
          000015          X=X+1
;*****
;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 1 (RANDOM STALLS AFTER
;PUNCHING EACH CHARACTER. MAXIMUM STALL 47 MILLISECONDS)
007672 012737 177720 004242 P1NA:  MOV  #177720,STLMSK ;SET STALL MASK FOR 57(8) MAX
007700 012737 001000 001330 P1NA:  MOV  #512.,RCNT  ;SET CHARACTER COUNT TO 512.
007706 004537 010122          JSR   %5,PFRNT  ;GO PUNCH FRONT END, AND MODE 1
007712 000001          1          ;INDICATOR
007714 104026          INBIN          ;INITIALIZE SPECIAL BINARY COUNT.
007716 104030          P1NB:  GETBNP          ;GET BINARY CHARACTER.
007720 004737 004572          JSR   %7,HSPCH  ;GO PUNCH THE CHARACTER.
007724 104012          STALL          ;RANDOM STALL.
007726 005337 001330          DEC   RCNT      ;DECREMENT CHAR COUNT.
007732 001371          BNE   P1NB      ;BRANCH IF COUNT NOT YET 0.
007734 104000          SCOPE          ;SCOPE
007736          TSTA  P10,5
007736          TSTAA P10,5,\X+1,\X+2,\Y
;*****
007736 000016          P1T16: 16          ; PRG1 ROUTINE 16          *
007740 010034          P1T17          ;ADDRESS OF NEXT ROUTINE          *
007742 000005          5          ;TEST ITERATION COUNT          *
007744 007762          P10A          ;SCOPE ENTRY POINT          *
          000016          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)
007746 012737 177720 004242 P10A:  MOV  #177720,STLMSK ;SET STALL MASK FOR 57(8) MAX.
007754 012737 177760 004270 P10A:  MOV  #177760,RCMSK ;SET CHAR GROUP MASK FOR 17(8) MAX).
007762 012737 001000 001330 P10A:  MOV  #512.,RCNT  ;SET CHARACTER COUNT TO 512.
007770 004537 010122          JSR   %5,PFRNT  ;GO PUNCH FRONT END AND MODE 2
007774 000002          2          ;INDICATOR
007776 104026          INBIN          ;INITIALIZE SPECIAL BINARY COUNT.
010000 004737 004244          P10B:  JSR   %7,GRcnt ;GENERATE RANDOM CHARACTER COUNT
010004 104012          STALL          ;RANDOM STALL.
```

```

010006 104030          P10C:  GETBNP          ;GET BINARY CHARACTER.
010010 004737 004572   JSR      %7,HSPCH      ;PUNCH THE CHARACTER.
010014 005337 001330   DEC      RCNT          ;DECREMENT CHAR COUNT
010020 001404          BEQ      P10D          ;BRANCH IF COUNT IS 0.
010022 005337 004272   DEC      RNCNT         ;NOT 0. DECREMENT RANDOM CHAR COUNT.
010026 001367          BNE      P10C          ;BRANCH IF COUNT NOT YET 0.
010030 000763          BR       P10B          ;BRANCH IF COUNT 0.
010032 104000          P10D:  SCOPE          ;SCOPE
010034          TSTAA   P1P,1,\X+1,LST,\Y

;*****
P1T17:  17              ; PRG1 ROUTINE 17 *
010036 177777          P1TLST         ;ADDRESS OF NEXT ROUTINE *
010040 000001          1              ;TEST ITERATION COUNT *
010042 010044          P1PA          ;SCOPE ENTRY POINT *
          000017          X=X+1

;*****
;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 3.
;STALL 10 SECONDS, PUNCH 32 CHARACTERS, UNTIL THE ENTIRE PATTERN IS
;COMPLETED.
010044 012737 001000 001330 P1PA:  MOV      #512.,RCNT      ;SET CHARACTER COUNT TO 512.
010052 004537 C10122      JSR      %5,PFRNT      ;GO PUNCH FRONT END AND MODE 3
010056 000003          3              ;INDICATOR.
010060 104026          INBIN          ;INITIALIZE SPECIAL BIN COUNT
010062 104031          P1PB:  DELAY          ;STALL 10 SECONDS
010064 023420          10000.
010066 012737 000040 004272 P1PC:  MOV      #32.,RNCNT      ;SET GROUP COUNT TO 32.
010074 104030          P1PC:  GETBNP          ;GET BINARY CHARACTER
010076 004737 004572   JSR      %7,HSPCH      ;PUNCH CHARACTER
010102 005337 001330   DEC      RCNT          ;DECREMENT CHAR COUNT
010106 001404          BEQ      P1PD          ;BRANCH IF COUNT IS 0
010110 005337 004272   DEC      RNCNT         ;DECREMENT GROUP COUNT
010114 001367          BNE      P1PC          ;BRANCH IF COUNT NOT YET 0.
010116 000761          BR       P1PB          ;BRANCH IF COUNT 0.
010120 104000          P1PD:  SCOPE          ;SCOPE

;SUBROUTINE TO PUNCH FRONT END AND MODE CODE (USED BY PRG3).
PFRNT:  MOV      #20.,%1      ;PUNCH 20 BLANK CHARACTERS (000)
          CLR      PCHOUT      ;CLEAR PCHOUT.
          JSR      %7,HSPCH      ;PUNCH CHAR.
          DEC      %1          ;DECREMENT R1
          BNE      -6          ;BRANCH IF NOT 20 CHARCTERS YET.
          MOV      #377,PCHOUT ;PUNCH RUBOUT CHAR (SYNC CHAR).
          JSR      %7,HSPCH
          MOV      (5)+,PCHOUT ;MOVE MODE CODE TO PCHOUT
          JSR      %7,HSPCH      ;PUNCH MODE CODE.
          MOV      #4,%1        ;PUNCH 4 BLANK CHARACTERS.
          CLR      PCHOUT
          JSR      %7,HSPCH
          DEC      %1
          BNE      -6
          RTS      %5          ;EXIT

;PRG2 - PUNCH VERIFY PROGRAM
;THIS PROGRAM VERIFIES TAPE PRODUCED BY PRG1.
;ANY ERRORS FOUND ARE REPORTED.
010206 104010          PRG2:  TYPE          ;TYPE TITLE
  
```

```

010210 013514          TITL2
010212 004737 005020 JSR      PC,RDRSEL      ;SELECT READER.
010216 104011          TYPES
010220 013547          IM2
010222 013243          IM6
010224 014041          IM23
010226 177777          -1
010230 104024          CHALT
010232 012737 000372 001352 ETOA:  MOV    #250.,CTRA      ;WAIT FOR USER.
010240 012737 000012 001354 ETOB:  MOV    #10.,CTRB      ;250 TO CTRA.(TOTAL CHAR COUNT).
010246 004737 004064          ETOC:  JSR    %7,BREAD      ;READ CHAR
010252 005737 001334          TST    CRBUF
010256 001007          BNE    ETOD      ;BRANCH IF NON-ZERO CHAR.
010260 005337 001354          DEC    CTRB      ;0 CHAR. DECREMENT CTRB
010264 001413          BEQ    ETOF      ;BRANCH IF 10 CONSFUTIVE 0'S READ.
010266 005337 001352          DEC    CTRA      ;NO. DECREMENT CTRA.
010272 001365          BNE    ETOC      ;BRANCH IF NOT YET 250 CHARS READ.
010274 000403          BR     ETOE      ;250 CHARS READ. SYNE ERROR.
010276 005337 001352          ETOD:  DEC    CTRA      ;DECREMENT CTRA
010302 001356          BNE    ETOB      ;BRANCH IF NOT 250 CHARS READ YET.
010304 104016          ETOE:  ERRORN
010306 015220          EM3
010310 177777          -1
010312 000747          BR     ETOA      ;GO TRY AGAIN.
010314 004737 004064          ETOF:  JSR    %7,BREAD      ;READ CHAR
010320 005737 001334          TST    CRBUF
010324 001004          BNE    ETOG      ;BRANCH IF NON-ZERO CHAR.
010326 005337 001352          DEC    CTRA      ;DECREMENT CTRA
010332 001370          BNE    ETOF      ;BRANCH IF NOT 250 CHARS READ YET.
010334 000763          BR     ETOE      ;250 CHARS READ. SYNC ERROR.
010336 012737 000377 001326 ETOG:  MOV    #377,ERRT
010344 043737 001350 001326 BIC    RDRMSK,ERRT
010352 023737 001326 001334 CMP    ERRT,CRBUF      ;COMPARE CHAR READ TO 377.
010360 001414          BEQ    ETOH      ;377.OK.
010362 104017          OACNV
010364 001326          ERRT
010366 015262          ESB
010370 000004          4
010372 104017          OACNV
010374 001334          CRBUF
010376 015275          EWAS
010400 000004          4
010402 104016          ERRORN
010404 015235          EM4
010406 177777          -1
010410 000710          BR     ETOA      ;START OVER
010412 004737 004064          ETOH:  JSR    %7,BREAD      ;READ CHAR.
010416 023727 001334 000003 CMP    CRBUF,#3
010424 101410          BLOS  ETOI      ;COMPARE CHAR READ TO 3.
010426 104017          OACNV      ;BRANCH IF SAME OR LOWER.
010430 001334          CRBUF      ;ERROR. CONVERT DATA READ TO ASCII.
010432 015354          FWAS      ;SET UP FOR TYPEOUT.
010434 000004          4
010436 104016          ERRORN
010440 015302          EM5
010442 177777          -1

```

```

010444 000672          BR      ETOA          :START OVER.
010446 012737 000004 001352 ETOI:  MOV      #4,CTRA      :4 TO CTRA (CHAR COUNT)
010454 005037 003212          CLR      BINR        :CLEAR BINR. EXPECTED CHAR IS 0.
010460 004737 004064          ETOJ:  JSR      %7,BREAD    :READ CHAR.
010464 004737 010530          JSR      %7,ECHK     :GO CHECK CHAR READ.
010470 005337 001352          DEC      CTRA        :DECREMENT CTRA
010474 001371          BNE      ETOJ        :BRANCH IF NOT 4 CHARS READ YET.
010476 104026          INBIN          :INITIALIZE SPECIAL BINARY COUNT.
010500 012737 001000 001352 ETOK:  MOV      #512,CTRA   :SET CHAR COUNT TO 512.
010506 004737 004064          JSR      %7,BREAD    :READ CHAR.
010512 104027          GETBNR         :GET BIN CHAR.
010514 004737 010530          JSR      %7,ECHK     :GO CHECK CHAR READ.
010520 005337 001352          DEC      CTRA        :DECREMENT CHAR COUNT
010524 001370          BNE      ETOK        :BRANCH IF NOT 512 CHARS READ YET.
010526 000641          BR      ETOA        :DONE. START OVER.
010530 023737 001334 003212 ECHK:  CMP      CRBUF,BINR  :COMPARE CHAR READ AGAINST EXPECTED CHAR.
010536 001413          BEQ      ECHKA      :BRANCH IF EQUAL.
010540 104017          OACNV         :CONVERT EXPECTED DATA TO ASCII.
010542 003212          BINR
010544 015126          ASB
010546 000004          4
010550 104017          OACNV          :CONVERT DATA READ TO ASCII.
010552 001334          CRBUF
010554 015141          AWAS
010556 000004          4
010560 104016          ERRORN        :ERROR. DATA ERROR.
010562 015103          EM1
010564 177777          -1
010566 000207          ECHKA: RTS      %7          :EXIT
:PRG3 - COMBINED READER-PUNCH TEST
:USES SPECIAL BINARY COUNT PATTERN.
PRG3:  TYPE          :TYPE TITLE.
010570 104010          TITL3
010572 014505          JSR      PC,RDRSEI   :SELECT READER
010574 004737 005020          JSR      PC,PCHSEL   :SELECT PUNCH.
010600 004737 004622          TYPES
010604 104011          IM3
010606 014551          IM6
010610 013243          IM23
010612 014041          -1
010614 177777          CHALT
010616 104024          INBIN          :INITIALIZE BINARY COUNTS.
010620 104026          MOV      #177620,STLMSK :SET MAX. STALL DELAY.
010622 012737 177620 004242 CLR      PCHCNT      :CLEAR PUNCH COUNT
010630 005037 011124          CLR      RBUSY      :CLEAR READER BUSY INDICATOR
010634 005037 011126          STRDRV         :SET PTRI VECTOR.
010640 104013          WNZERO
010642 011130          STPCHV        :SET PTPI VECTOR.
010644 104014          PBIN
010646 010676          GETBNP        :GET BIN CHARACTER
010650 104030          MOV      BINP,@PPB   :PUNCH IT
010652 013777 003214 170326 BIS      #BIT6,@PPS   :ENABLE PTPI
010660 052777 000100 170316 CLR      PSW         :SET PRIORITY 0.
010666 005037 177776          TRAP         :TRAP CALL TO CAUSE NOISE.
010672 104400          BR      -2
010674 000776          BR      @PPS
010676 005777 170302          PBIN:  TST          :TEST FOR ERROR.
  
```

010702	100004			BPL	PBNA				:BRANCH IF NO ERROR.
010704	104010			TYPE					:TYPE PUNCH NOT READY
010706	014707			SM3					:MESSAGE.
010710	104024			CHALT					
010712	000771			BR	PBIN				:RECHECK FOR ERROR.
010714	105777	170264		TSTB	@PPS				:CHECK FOR DONE.
010720	100403			BMI	PBNB				:BRANCH IF DONE SET.
010722	104016			ERRORN					:ERROR.FALSE PUNCH INTERRUPT.
010724	015451			EM11					
010726	177777			-1					
010730	005237	011124		INC	PCHCNT				:INCREMENT PUNCH COUNT.
010734	104030			GETBNP					:GET BINARY CHARACTER
010736	013777	003214	170242	MOV	BINP,@PPB				:ENABLE PUNCH
010744	023727	011124	000144	CMP	PCHCNT,#100.				:NOT BUSY. PUNCH COUNT 100 YET?
010752	103001			BHIS	+.4				:BRANCH IF YES..
010754	000002			RTI					:NOT YET. EXIT INTERRUPT
010756	105737	011126		TSTB	RBUSY				:READER BUSY?
010762	100406			BMI	PBNC				:BR IF YES.
010764	052737	000200	011126	BIS	#BIT7,RBUSY				:SET READER BUSY
010772	052777	000101	170200	BIS	#101,@PRS				:ENABLE PTRI AND READER.
011000	023727	011124	000146	PBNC: CMP	PCHCNT,#102.				:PUNCH COUNT 102?
011006	101402			BLOS	PBND				:BR IF NOT.
011010	005077	170170		CLR	@PPS				:STOP PUNCH.
011014	104012			PBND: STALL					
011016	000002			RTI					:EXIT INTERRUPT.
011020	005777	170154		CREAD: TST	@PRS				:CHECK FOR ERROR.
011024	100003			BPL	CRDA				:BRANCH IF NO ERROR.
011026	004737	003772		JSR	%7,TSM2				:ERROR.TYPE MESSAGE.
011032	000772			BR	CREAD				
011034	105777	170140		CRDA: TSTB	@PRS				:TEST FOR DONE.
011040	100403			BMI	CRDAA				:BRANCH IF DONE SET.
011042	104016			ERRORN					:ERROR.FALSE READER INTERRUPT.
011044	015421			EM10					
011046	177777			-1					
011050	017737	170126	001334	CRDAA: MOV	@PRB,CRBUF				:CHARACTER READ TO CRBUF
011056	005337	011124		DEC	PCHCNT				
011062	001015			BNE	CREADC				:BR IF NOT 0.
011064	032777	000100	170112	BIT	#BIT6,@PPS				:NO. PTPI ENABLED?
011072	001003			BNE	CREADA				:BR IF YES.
011074	052777	000100	170102	BIS	#BIT6,@PPS				:NO. ENABLE PTPI.
011102	04737	000200	011126	CREADA: BIC	#BIT7,RBUSY				:YES. CLEAR READER BUSY.
011110	005077	170064		CLR	@PRS				:DISABLE PTRI.
011114	000207			RTS	%7				:EXIT.
011116	005277	170056		CREADC: INC	@PRS				:ENABLE READER
011122	000207			RTS	%7				:EXIT.
011124	000000			PCHCNT: OPEN					
011126	000000			RBUSY: OPEN					
011130	004737	011020		WNZERO: JSR	%7,CREAD				:READ CHARACTER
011134	005737	001334		TST	CRBUF				:CHECK CHARACTER
011140	001001			BNE	+.4				:BRANCH IF NON-ZERO CHAR.
011142	000002			RTI					:ZERO. EXIT INTERRUPT.
011144	012777	011162	170036	MOV	#RBIN,@RDRVTR				:SET READER VECTOR TO READ BINARY
011152	012737	000003	001344	MOV	#3,ERCTR				:COUNT. SET ERROR COUNTER TO 3.
011160	000402			BR	RBINA				
011162	004737	011020		RBIN: JSR	%7,CREAD				:READ CHARACTER.
011166	104027			RBINA: GETBNR					:GET BINARY CHARACTER

```

011170 023737 003212 001334      CMP      BINR,CRBUF      ;COMPARE AGAINST CHAR READ.
011176 001001                    BNE      RBINB          ;BRANCH IF NOT SAME.
011200 000002                    RTI                          ;SAME EXIT INTERRUPT.
011202 104017                    RBINB: OACNV           ;CONVERT EXPECTED CHAR TO ASCII
011204 003212                    BINR
011206 015126                    ASB
011210 000004                    4
011212 104017                    OACNV           ;CONVERT RECEIVED CHAR TO ASCII
011214 001334                    CRBUF
011216 015141                    AWAS
011220 000004                    4
011222 104016                    ERRORN          ;ERROR MESSAGE. DATA ERROR.
011224 015103                    EM1
011226 177777                    -1
011230 005337 001344            DEC      ERCTR          ;3RD ERROR?
011234 001001                    BNE      RBINC          ;YES.
011236 000002                    RTI                          ;NO. EXIT INTERRUPT.
011240 052737 100000 011126    RBINC: BIS      #BIT15,RBUSY ;DISABLE STALLS.
011246 012777 011270 167734    MOV      #RBIND,@RDRVTR ;SET PTR VECTOR TO RBIND.
011254 012737 000004 001344    MOV      #4,ERCTR       ;USE ERCTR AS CHARACTER COUNTER.
011262 012700 001334          MOV      #CRBUF,%0      ;ADDR OF CRBUF TO %0
011266 000002                    RTI                          ;EXIT INTERRUPT
011270 004737 011020          RBIND: JSR      %7,CREAD  ;READ CHARACTER
011274 013720 001334          MOV      CRBUF,(0)+     ;STORE CHARACTER STARTING AT CHR1
011300 005337 001344          DEC      ERCTR          ;3RD CHARACTER?
011304 001401                    BEQ      .+4             ;YES.
011306 000002                    RTI                          ;EXIT INTERRUPT. NOT 3RD YET.
011310 004737 004440          JSR      %7,SYNCA       ;GO SYNC THE READER.
011314 000751                    BR       RBINC           ;NO SYNC. TRY AGAIN.
011316 012777 011162 167664    MOV      #RBIN,@RDRVTR ;SYNCED. SET READER VECTOR TO RBIN.
011324 012737 000003 001344    MOV      #3,ERCTR       ;SET ERROR COUNT TO 3.
011332 042737 100000 011126    BIC      #BIT15,RBUSY   ;ENABLE STALLS.
011340 000002                    RTI                          ;EXIT INTERRUPT.
                                ;PRG4 - PUNCHES 2 CHARACTERS SET IN SR.
011342 104010                    PRG4: TYPE              ;TYPE TITLE.
011344 013620                    TITL4
011346 004737 004622          JSR      PC,PCHSEL      ;SELECT PUNCH.
011352 104011                    TYPES
011354 013663                    IM4
011356 013314                    IM16
011360 014041                    IM23
011362 177777                    -1
011364 104024                    CHALT            ;COMMON HALT. WAIT FOR USER.
011366 113737 177570 001332    PRG4A: MOVB      SR,PCHOUT ;PUNCH FIRST CHARACTER.
011374 004737 004572          JSR      %7,HSPCH
011400 113737 177571 001332    MOVB      SR+1,PCHOUT   ;PUNCH SECOND CHARACTER.
011406 004737 004572          JSR      %7,HSPCH
011412 000765                    BR       PRG4A          ;REPEAT.
                                ;PRG5 - READS-CHECKS TAPE PUNCHED WITH CODES SET IN SR
011414 104010                    PRG5: TYPE              ;TYPE TITLE.
011416 013723                    TITL5
011420 004737 005020          JSR      PC,RDRSEL      ;SELECT READER.
011424 104011                    TYPES
011426 014004                    IM5
011430 013243                    IM6
011432 014041                    IM23
  
```


011434	177777				-1			
011436	104024				CHALT			:COMMON HALT. WAIT FOR USER.
011440	113737	177570	001356		MOVB	SR,CTRC		:STORE EXPECTED CHARACTERS.
011446	113737	177571	001360		MOVB	SR+1,CTRD		:IN CTRC AND CTRD.
011454	004737	004064		HTOA:	JSR	%7,BREAD		:MATCH CHARS ON TAPE AGAINST EXPECTED CHARS.
011460	013737	001334	001336		MOV	CRBUF,CHR1		:READ CHAR INTO CHR1
011466	004737	004064			JSR	%7,BREAD		:READ CHAR
011472	013737	001334	001340		MOV	CRBUF,CHR2		:INTO CHR2
011500	023737	001336	001356		CMP	CHR1,CTRC		:(CHR1)=(CTRC)?
011506	001041				BNE	HTOE		:NO.
011510	023737	001340	001360		CMP	CHR2,CTRD		:YES. (CHR2)=(CTRD)?
011516	001062				BNE	HTOG		:NO. MATCH ERROR.
011520	005037	001354			CLR	CTRB		:YES. NEXT CHAR SHOULD = (CTRC) (CTRB-0)
011524	012737	000003	001344	HTOB:	MOV	#3,ERCTR		:3 TO ERROR COUNTER.
011532	004737	004064		HTOC:	JSR	%7,BREAD		:READ CHAR
011536	005137	001354			COM	CTRB		:COMPLEMENT CHAR INDICATOR
011542	001437				BEQ	HTOF		:BRANCH IF EXPECTED CHAR SHOULD = (CTRD)
011544	023737	001334	001356		CMP	CRBUF,CTRC		:(CRBUF) = (CTRC)?
011552	001767				BEQ	HTOC		:YES.
011554	104017				OACNV			:NO. (CTRC) TO ASB IN ASCII FORM.
011556	001356				CTRC			
011560	015126				ASB			
011562	000004				4			
011564	104017			HTOD:	OACNV			:(CRBUF) TO AWAS IN ASCII FORM.
011566	001334				CRBUF			
011570	015141				AWAS			
011572	000004				4			
011574	104016				ERRORN			:ERROR 1 CALL. TYPE DATA ERROR MESSAGE.
011576	015103				EM1			
011600	177777				-1			
011602	005337	001344			DEC	ERCTR		:3 ERRORS?
011606	001722				BEQ	HTOA		:YES. START ALL OVER.
011610	000750				BR	HTOC		:NO. CONTINUE READING.
011612	023737	001336	001360	HTOE:	CMP	CHR1,CTRD		:(CHR1) = (CTRD)?
011620	001021				BNE	HTOG		:NO. MATCH ERROR.
011622	023737	001340	001356		CMP	CHR2,CTRC		:YES. (CHR2) = (CTRC)?
011630	001015				BNE	HTOG		:NO. MATCH ERROR.
011632	012737	177777	001354		MOV	#-1,CTRB		:YES. NEXT CHAR SHOULD = (CTRD)
011640	000731				BR	HTOB		:GO START READING.
011642	023737	001334	001360	HTOF:	CMP	CRBUF,CTRD		:(CRBUF) = (CTRD)?
011650	001730				BEQ	HTOC		:YES. OK. CONTINUE READING.
011652	104017				OACNV			:NO. (CTRD) TO ASB IN ASCII FORM.
011654	001360				CTRD			
011656	015126				ASB			
011660	000004				4			
011662	000740				BR	HTOD		:GO GENERATE ERROR MESSAGE.
011664	104016			HTOG:	ERRORN			:MATCH ERROR. UNABLE TO MATCH UP
011666	015361				EM6			:2 CONSECUTIVE CHARACTERS FROM READER
011670	177777				-1			
011672	000670				BR	HTOA		:TO CHARACTERS READ FROM SR.
011674	104010				:PRG6 -	READ X CHARACTERS, STALL Y MILLISECONDS.		
011676	013337			PRG6:	TYPE			:TYPE TITLE AND INSTRUCTIONS.
011700	004737	005020			TITL6			
011704	104010				JSR	PC,RDRSEL		:SELECT READER.
					TYPE			

```

011706 013371          IM17
011710 104024          CHALT
011712 005037 011754   ITA:  CLR      ITY
011716 005037 011760   CLR      ITX
011722 113737 177570 011754   MOVB    SR,ITY      ;MOVE STALL COUNT TO ITY.
011730 113737 177571 011760   MOVB    SR+1,ITX    ;MOVE CHAR COUNT TO ITX.
011736 001405          BEQ      ITC          ;BR IF COUNT 0.
011740 004737 003714   ITB:  JSR      %7,AREAD ;FETCH CHARACTER.
011744 105337 011760   DECB    ITX          ;DECREMENT CHAR COUNT.
011750 001373          BNE      ITB          ;BRANCH IF COUNT NOT 0.
011752 104031          ITC:  DELAY
011754 000000          ITY:  OPEN          ;READ CHARS. STALL NOW.
011756 000755          BR       ITA          ;STALL COUNT IN MSECS.
011760 000000          ITX:  OPEN          ;REPEAT
;PRG7. PUNCH SPECIAL BINARY COUNT PATTERN TEST TAPE
PRG7:  TYPE
      TITL7          ;TYPE TITLE.
011762 104010          JSR      PC,PCHSEL    ;SELECT PUNCH.
011764 013175          TYPES
011766 004737 004622   IM16
011772 104011          -1
011774 013314          CHALT
011776 177777          ;WAIT FOR USER.
012000 104024          MOV      #20.,-(6)   ;PUNCH 20 BLANK CHAR. LEADER
012002 012746 000024   CLR      PCHOUT
012006 005037 001332   PRG7A: JSR      %7,HSPCH
012012 004737 004572   DEC      @%6
012016 005316          BNE      PRG7A
012020 001374          INBIN          ;INITIALIZE SPECIAL BINARY COUNT
012022 104026          PRG7B: GETBNP      ;GET BINARY CHARACTER.
012024 104030          JSR      %7,HSPCH    ;PUNCH CHARACTER
012026 004737 004572   BR       PRG7B      ;REPEAT.
012032 000774          ;PRG10 - READER SPEED PRINT LOOP
PRG10: TYPE
      TITL10         ;TYPE TITLE
012034 104010          JSR      PC,RDRSEL    ;SELECT READER.
012036 014062          TYPES
012040 004737 005020   IM10
012044 104011          IM24A
012046 014113          -1
012050 014302          CHALT
012052 177777          ;HALT. WAIT FOR USER.
012054 104024          KTA:  CLR      CTRC   ;CLEAR WORK REGISTERS
012056 005037 001356   CLR      CTRB
012062 005037 001354   BIT      #BIT14,SR   ;DETERMINE WHETHER 30 OR
012066 032737 040000 177570 BEQ      KTB          ;300 SECOND TIMING IS DESIRED
012074 001403          MOV      #270.,CTRC ;SET UP FOR DESIRED TIME BASE.
012076 012737 000416 001356 KTB:  ADD      #30.,CTRC
012104 062737 000036 001356 BR       KTD
012112 000407          KTC:  JSR      %7,BREAD ;READ CHARACTER.
012114 004737 004064   DEC      CTRA        ;DECREMENT CTRA
012120 005337 001352   BNE      KTE          ;BRANCH IF CTRA NOT 0.
012124 001005          INC      CTRB        ;CTRA0.+1 TO CTRB.
012126 005237 001354   MOV      CTRC,CTRA   ;RELOAD CTRA.
012132 013737 001356 001352 KTD:  TST      SR       ;TIME UP?
012140 005737 177570   KTE:  BPL      KTC     ;NO.
012144 100363          KTF:  JSR      %5,CPKPL ;GO TYPE OUT DEVICE SPEED.
012146 004537 012254   SM4
012152 014732

```

```

012154 000737          BR      KTA-2          ;GO HALT.
;PRG11 - PUNCH SPEED PRINT LOOP
012156 104010          PRG11: TYPE          ;TYPE TITLE.
012160 014454          TITL11
012162 004737 004622  JSR      PC,PCHSEL      ;SELECT PUNCH.
012166 104011          TYPES
012170 013314          IM16
012172 014302          IM24A
012174 177777          -1
012176 104024          CHALT          ;HALT. WAIT FOR USER.
012200 005037 001354  LTA:  CLR      CTRB          ;CLEAR WORK AREAS.
012204 005037 001332  CLR      PCHOUT
012210 000407          BR      LTC
012212 004737 004572  LTB:  JSR      %7,HSPCH      ;PUNCH A 0
012216 005337 001352  DEC      CTRA          ;DECREMENT CTRA
012222 001005          BNE      LTD          ;BRANCH IF CTRA NOT 0
012224 005237 001354  INC      CTRB          ;INCREMENT CTRB.
012230 012737 000074 001352 LTC:  MOV      #60.,CTRA      ;MOVE 60 TO CTRA
012236 005737 177570 LTD:  TST      SR          ;TIME UP?
012242 100363          BPL      LTB          ;NO.
012244 004537 012254 LTE:  JSR      %5,CPKPL      ;GO TYPE OUT DEVICE SPEED.
012250 014754          SMS
012252 000751          BR      LTA-2
012254 012537 012272 CPKPL: MOV      (5)+,CPKPLA      ;GO HALT AND READY UP FOR NEXT TIME.
012260 104020          BDCNV          ;MOVE ADDR OF 1ST MESSAGE TO CPKPLA.
012262 001354          CTRB          ;CONVERT (CTRB) TO DECIMAL ASCII.
012264 014775          ACPS
012266 000003          3
012270 104011          TYPES          ;TYPE DEVICE SPEED.
012272 000000          CPKPLA: OPEN
012274 014775          ACPS
012276 177777          -1
012300 000205          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.

012302 177777
 012304 000000

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

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

012306 104010
 012310 015705
 012312 004737 004622
 012316 104011
 012320 013314
 012322 014041
 012324 177777
 012326 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.

012330 042777 000006 166646
 012336 004737 012444
 012342 113737 012302 001332
 012350 004737 012372
 012354 113737 012304 001332
 012362 004737 012372
 012366 000137 012342

PG12B: BIC #6,@PPS ;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

012372 000240
 012374 043737 001346 001332
 012402 013777 001332 166576
 012410 004737 012476
 012414 000000
 012416 052777 000400 166560
 012424 032737 040000 177570
 012432 001371

HSPCH1: NOP ;DEBUG.
 :PUNCH THE CHARACTER
 BIC PCHMSK,PCHOUT
 MOV PCHOUT,@PPB ;LOAD PUNCH BUFFER.
 ;WAIT FOR DONE
 JSR PC,TIMER ;IN A TIMEOUT LOOP.
 HALT ;TIMED-OUT RETURN
 1\$: BIS #400,@PPS ;RE-INIT PUNCH.
 BIT #BIT14,SR ;SCOPE LOOP ENABLED???
 BNE 1\$;YES,LOOP UNTIL SR14 0.

012434 004737 012476 JSR PC, TIMER ;RE-CHECK THE DONE BIT.
012440 000000 HALT ;FATAL ERROR: DONE BIT WAS NO
;RESET BY THE INIT INSTRUCTION.
;ELSE RETURN IS TO THE NEXT INSTRUCTION.

012442 000207 ;OUTPUT ERROR MESSAGE OF PUNCH NOT READY
RTS PC ;EXIT

;IF ERROR BIT 15 SET OR BIT 7 NOT SET.

012444 005777 166534 TSTPUN: TST @PPS
012450 100404 BMI HSPCH2
012452 105777 166526 TSTB @PPS
012456 100001 BPL HSPCH2
012460 000207 RTS PC
012462 104011 HSPCH2: TYPES
012464 014707 SM3 ;'PUNCH NOT READY'
012466 013314 IM16 ;'MAKE PUNCH READY'
012470 177777 -1 ;TERMINATOR
012472 104024 CHALT
012474 000763 BR TSTPUN ;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+1

012476 012737 177200 001354 TIMER: MOV #-600,CTRB ;MAXIMUM=600 MS DELAY
012504 000240 TIMER1: NOP
012506 005237 001354 INC CTRB ;UP COUNTER
012512 001406 BEQ TIMER2 ;TIMED-OUT...ERROR..
012514 105777 166464 TSTB @PPS ;READY BIT SET?
012520 100405 BMI TIMER3 ;YES, OK

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

012522 104031 DELAY
012524 000012 10. ;10 MILLI SE DELAY.
012526 000766 BR TIMER1 ;GO CHECK AGAIN.

012530 000240 TIMER2: NOP
012532 000207 RTS PC ;RETURN TO CALL+1

012534 062716 000002 TIMER3: ADD #2,(SP) ;RETURN TO CALL+2 IF NO ERROR
012540 000207 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.

;

012542	104010			PRG13:	TYPE				;OUTPUT TEST HEADER.
012544	015753				TITL13				;TEXT 'PRG13--READER LIGHT TEST'
012546	004737	005020		PRG13C:	JSR	PC,RDRSEL			;SELECT READER VIA SWR.
012552	104011			PRG13B:	TYPES				;OUTPUT MESSAGE:
012554	016005				IM13A				;TEXT 'TURN LIGHT ON VIA RDR SWITCH'
012556	016066				IM13B				;TEXT 'USE SW8 TO TURN LIGHT
012560	177777				-1				; OFF'
012562	013700	177570							
012566	032700	000400							
012572	001773			PG13A:	MOV	SR,R0			;CHECK BIT 8.
012574	052777	000400	166376		BIT	#400,R0			;SWITCH DOWN, NO ACTION REQUIRED.
012602	032737	040000	177570	1\$:	BEQ	PG13A			;ELSE TURN LIGHT OFF.
012610	001371				BIS	#400,@PRS			; IS SCOPE LOOP SET?
012612	000000				BIT	#BIT14,SR			;YES, LOOP UNTIL SR14=0.
012614	032737	010000	177570		BNE	1\$;WAIT FOR OPR TO HIT CONTINUE
012622	001351				HALT				;CHECK FOR SELECT NEW READER.
012624	000752				BIT	#10000,SR			;YES, SELECT NEW READER.
					BNE	PRG13C			
					BR	PRG13B			

PA611 MACY11 30A(1052) 14-JUN-78 14:20 PAGE 41
CZPAAC.P11 13-JUN-78 12:38

K 5

SEQ 0062

012626 022445 120

APGEND: .ASCII '%%P'

012631	040	020040	047105	APN:	.ASCII	' END.a'
012636	027104	100				
012641	045	037445	047111	CM2:	.ASCII	'%%?INVALID PROGRAM.a'
012646	040526	044514	020104			
012654	051120	043517	040522			
012662	040115					
012664	022445	044477	053116	CM3:	.ASCII	'%%?INVALID TEST.a'
012672	046101	042111	052040			
012700	051505	040124				
012704	022445	042523	020124	ASETSR:	.ASCII	'%%SET SR OPTIONS. NORMAL SR IS 000000.a'
012712	051123	047440	052120			
012720	047511	051516	020056			
012726	047516	046522	046101			
012734	051440	020122	051511			
012742	030040	030060	030060			
012750	040060					
012752	022445	054524	042520	PGTIT:	.ASCII	'%%TYPESET 11 READER-PUNCH TESTS.a'
012760	042523	020124	030461			
012766	051040	040505	042504			
012774	026522	052520	041516			
013002	020110	042524	052123			
013010	022523	100				
013013	045	042522	052123	RUNINS:	.ASCII	'%RESTART PROGRAM.a'
013020	051101	020124	051120			
013026	043517	040522	027115			
013034	100					
013035	045	042523	020124	MSVCTR:	.ASCII	'%SET SR WITH RDRO VECTOR.a'
013042	051123	053440	052111			
013050	020110	042122	030122			
013056	053040	041505	047524			
013064	027122	100				
013067	045	042523	020124	SELRDR:	.ASCII	'%SET # OF READERS IN SR.a'
013074	020043	043117	051040			
013102	040505	042504	051522			
013110	044440	020116	051123			
013116	040056					
013120	051445	052105	021440	SELPCH:	.ASCII	'%SET # OF PUNCHES IN SR.a'
013126	047440	020106	052520			
013134	041516	042510	020123			
013142	047111	051440	027122			
013150	100					
013151	045	050045	043522	TITL1:	.ASCII	'%%PRG1. PUNCH TEST.a'
013156	027061	050040	047125			
013164	044103	052040	051505			
013172	027124	100				
013175	045	050045	043522	TITL7:	.ASCII	'%%PRG7. COUNT PATTERN TAPE GENERATOR.a'
013202	027067	041440	052517			
013210	052116	050040	052101			
013216	042524	047122	052040			
013224	050101	020105	042507			
013232	042516	040522	047524			
013240	027122	100				
013243	045	040515	042513	IM6:	.ASCII	'%MAKE READER READY.a'
013250	051040	040505	042504			
013256	020122	042522	042101			
013264	027131	100				

013267	045	050045	043522	TITL0:	.ASCII	'%%PRG0. READER TEST.@'
013274	027060	051040	040505			
013302	042504	020122	042524			
013310	052123	040056				
013314	046445	045501	020105	IM16:	.ASCII	'%MAKE PUNCH READY.@'
013322	052520	041516	020110			
013330	042522	042101	027131			
013336	100					
013337	045	050045	043522	TITL6:	.ASCII	'%%PRG6 - READ X, STALL Y.@'
013344	020066	020055	042522			
013352	042101	054040	020054			
013360	052123	046101	020114			
013366	027131	100				
013371	045	042523	020124	IM17:	.ASCII	'%SET SR15 TO SR8 TO NO. OF CHARS TO READ.'
013376	051123	032461	052040			
013404	020117	051123	020070			
013412	047524	047040	027117			
013420	047440	020106	044103			
013426	051101	020123	047524			
013434	051040	040505	026104			
013442	051445	052105	051440		.ASCII	'%SET SR7 TO SR0 TO NO. OF MSECS TO STALL.@'
013450	033522	052040	020117			
013456	051123	020060	047524			
013464	047040	027117	047440			
013472	020106	051515	041505			
013500	020123	047524	051440			
013506	040524	046114	040056			
013514	022445	051120	031107	TITL2:	.ASCII	'%%PRG2. PUNCH VERIFY TEST.@'
013522	020056	052520	041516			
013530	020110	042526	044522			
013536	054506	052040	051505			
013544	027124	100				
013547	045	047514	042101	IM2:	.ASCII	'%LOAD READER WITH TAPE PRODUCED '
013554	051040	040505	042504			
013562	020122	044527	044124			
013570	052040	050101	020105			
013576	051120	042117	041525			
013604	042105	040				
013607	102	020131	051120		.ASCII	'BY PRG1.@'
013614	030507	040056				
013620	022445	051120	032107	TITL4:	.ASCII	'%%PRG4. PUNCHES 2 CODES SET IN SR.@'
013626	020056	052520	041516			
013634	042510	020123	020062			
013642	047503	042504	020123			
013650	042523	020124	047111			
013656	051440	027122	100			
013663	045	042523	020124	IM4:	.ASCII	'%SET CODES TO BE PUNCHED IN SR.@'
013670	047503	042504	020123			
013676	047524	041040	020105			
013704	052520	041516	042510			
013712	020104	047111	051440			
013720	027122	100				
013723	045	050045	043522	TITL5:	.ASCII	'%%PRG5. READS TAPE PUNCHED WITH CODES SET IN SR.@'
013730	027065	051040	040505			
013736	051504	052040	050101			
013744	020105	052520	041516			

013752	042510	020104	044527	
013760	044124	041440	042117	
013766	051505	051440	052105	
013774	044440	020116	051123	
014002	040056			
014004	051445	052105	041440	IM5: .ASCII '%SET CODES TO BE READ IN SR.@'
014012	042117	051505	052040	
014020	020117	042502	051040	
014026	040505	020104	047111	
014034	051440	027122	100	
014041	040	051120	051505	IM23: .ASCII ' PRESS CONTINUE.@'
014046	020123	047503	052116	
014054	047111	042525	040056	
014062	022445	051120	030507	TITL10: .ASCII '%PRG10. RDR SPEED TEST.@'
014070	027060	051040	051104	
014076	051440	042520	042105	
014104	052040	051505	027124	
014112	100			
014113	045	047514	042101	IM10: .ASCII '%LOAD ANY TAPE LOOP IN READER '
014120	040440	054516	052040	
014126	050101	020105	047514	
014134	050117	044440	020116	
014142	042522	042101	051105	
014150	040			
014151	101	042116	046440	.ASCII 'AND MAKE READY.'
014156	045501	020105	042522	
014164	042101	027131		

014170	051445	052105	051440
014176	030522	020064	047524
014204	040440	030440	043040
014212	051117	031440	030060
014220	051440	041505	040
014225	124	046511	047111
014232	026107	047440	020122
014240	042523	020124	052111
014246	052040	020117	020060
014254	047506	020122	030063
014262	040		

.ASCII '%SET SR14 TO A 1 FOR 300 SEC '

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

014263 123 041505 047117
014270 020104 044524 044515
014276 043516 040056

.ASCII 'SECOND TIMING.a'

014302	050045	042522	051523	IM24A:	.ASCII	'%PRESS CONTINUE TO START TIMING.'
014310	041440	047117	044524			
014316	052516	020105	047524			
014324	051440	040524	052122			
014332	052040	046511	047111			
014340	027107					
014342	051445	052105	051440		.ASCII	'%SET SR 15 TO A 1 AT END OF '
014350	020122	032461	052040			
014356	020117	020101	020061			
014364	052101	042440	042116			
014372	047440	020106				
014376	044524	044515	043516		.ASCII	'TIMING PERIOD TO OBTAIN DEVICE SPEED '
014404	050040	051105	047511			
014412	020104	047524	047440			
014420	052102	044501	020116			
014426	042504	044526	042503			
014434	051440	042520	042105			
014442	040					
014443	124	050131	047505		.ASCII	'TYPEOUT.@'
014450	052125	040056				
014454	022445	051120	030507	TITL11:	.ASCII	'%%PRG11. PCH SPEED TEST.@'
014462	027061	050040	044103			
014470	051440	042520	042105			
014476	052040	051505	027124			
014504	100					
014505	045	050045	043522	TITL3:	.ASCII	'%%PRG3. COMBINED READER-PUNCH TEST.@'
014512	027063	041440	046517			
014520	044502	042516	020104			
014526	042522	042101	051105			
014534	050055	047125	044103			
014542	052040	051505	027124			
014550	100					
014551	045	040515	042513	IM3:	.ASCII	'%MAKE PUNCH READY, PUNCH BLANK LEADER, '
014556	050040	047125	044103			
014564	051040	040505	054504			
014572	020054	052520	041516			
014600	020110	046102	047101			
014606	020113	042514	042101			
014614	051105	020054				
014620	047514	042101	051040		.ASCII	'LOAD READER.@'
014626	040505	042504	027122			
014634	100					
014635	045	042522	042101	SM1:	.ASCII	'%READER ERROR BIT SET.@'
014642	051105	042440	051122			
014650	051117	041040	052111			
014656	051440	052105	040056			
014664	051045	040505	042504	SM2:	.ASCII	'%READER NOT READY.@'
014672	020122	047516	020124			
014700	042522	042101	027131			
014706	100					

014707	045	050045	047125	SM3:	.ASCII	'%%PUNCH NOT READY.@'
014714	044103	047040	052117			
014722	051040	040505	054504			
014730	040056					
014732	022445	042522	042101	SM4:	.ASCII	'%%READER SPEED : @'
014740	051105	051440	042520			
014746	042105	035040	040040			
014754	022445	052520	041516	SM5:	.ASCII	'%%PUNCH SPEED : @'
014762	020110	050123	042505			
014770	020104	020072	100			
014775	040	020040	020040	ACPS:	.ASCII	' CHARS PER SEC.@'
015002	044103	051101	020123			
015010	042520	020122	042523			
015016	027103	100				
015021	045	050045		EMO:	.ASCII	'%P'
015024	020040	020040	124	APNUMB:	.ASCII	' T'
015031	040	020040	020040	ATNUMB:	.ASCII	' PC'
015036	041520					
015040	020040	020040	020040	APC:	.ASCII	' ICNT '
015046	020040	041511	052116			
015054	040					
015055	040	020040	020040	AICNT:	.ASCII	' .@'
015062	040056					
015064	020040	050106	020103	FPCMSG:	.ASCII	' FPC '
015072	020040	020040	020040	AFPC:	.ASCII	' %@'
015100	022440	100				
015103	040	042040	052101	EM1:	.ASCII	' DATA ERROR S/B: '
015110	020101	051105	047522			
015116	020122	051440	041057			
015124	020072					
015126	020040	020040	020040	ASB:	.ASCII	' WAS: '
015134	040527	035123	040			
015141	040	020040	040040	AWAS:	.ASCII	' @'
015146	051040	051105	040505	EM2:	.ASCII	' REREAD ERROR. 1ST READ: '
015154	020104	051105	047522			
015162	027122	020040	051461			
015170	020124	042522	042101			
015176	020072					
015200	020040	020040	020040	ORGRD:	.ASCII	' WAS: '
015206	040527	035123	040			
015213	040	020040	040040	SUBRD:	.ASCII	' @'
015220	051440	047131	020103	EM3:	.ASCII	' SYNC ERROR.@'
015226	051105	047522	027122			
015234	100					
015235	045	042514	042101	EM4:	.ASCII	'%LEADER ERROR. S/B: '
015242	051105	042440	051122			
015250	051117	020056	051440			
015256	041057	020072				
015262	020040	020040	020040	ESB:	.ASCII	' WAS: '
015270	040527	035123	040			
015275	040	020040	040040	EWAS:	.ASCII	' @'
015302	046045	040505	042504	EM5:	.ASCII	'%LEADER ERROR. S/B BETWEEN '
015310	020122	051105	047522			
015316	027122	051440	041057			
015324	041040	052105	042527			
015332	047105	040				

015335	060	040440	042116		.ASCII	'0 AND 3. WAS : '
015342	031440	020056	040527			
015350	020123	020072				
015354	020040	020040	100	FWAS:	.ASCII	' @'
015361	040	040515	041524	EM6:	.ASCII	' MATCH ERROR.@'
015366	020110	051105	047522			
015374	027122	100				
015377	045	047516	051040	EM7:	.ASCII	'%NO RDR RESPONSE.@'
015404	051104	051040	051505			
015412	047520	051516	027105			
015420	100					
015421	040	040506	051514	EM10:	.ASCII	' FALSE READER INTERRUPT@'
015426	020105	042522	042101			
015434	051105	044440	052116			
015442	051105	052522	052120			
015450	100					
015451	040	040506	051514	EM11:	.ASCII	' FALSE PUNCH INTERRUPT@'
015456	020105	052520	041516			
015464	020110	047111	042524			
015472	051122	050125	040124			
015500	051445	052105	051040	SRDRM:	.ASCII	'%SET RDR # IN SR. SET SR15 IF 6 LEVEL.@'
015506	051104	021440	044440			
015514	020116	051123	020056			
015522	042523	020124	051123			
015530	032461	044440	020106			
015536	020066	042514	042526			
015544	027114	100				
015547	045	042523	020124	SPCHM:	.ASCII	'%SET PCH # IN SR. SET SR15 IF 6 LEVEL.@'
015554	041520	020110	020043			
015562	047111	051440	027122			
015570	051440	052105	051440			
015576	030522	020065	043111			
015604	033040	046040	053105			
015612	046105	040056				
015616	037445	047111	040526	INVRP:	.ASCII	'%?INVALID RDR/PCH.@'
015624	044514	020104	042122			
015632	027522	041520	027110			
015640	100					
015641	045	042122	020122	RDRIDM:	.ASCII	'%RDR '
015646	020040	051440	046105	ARDRID:	.ASCII	' SELECTED.@'
015654	041505	042524	027104			
015662	100					
015663	045	041520	020110	PCHIDM:	.ASCII	'%PCH '
015670	020040	051440	046105	APCHID:	.ASCII	' SELECTED.@'
015676	041505	042524	027104			
015704	100					
015705	045	050045	043522	TITL12:	.ASCII	'%PRG12--PROGRAMMABLE INIT PUNCH TEST@'
015712	031061	026455	051120			
015720	043517	040522	046515			
015726	041101	042514	044440			
015734	044516	020124	052520			
015742	041516	020110	042524			
015750	052123	100				
015753	045	051120	030507	TITL13:	.ASCII	'%PRG13--READER LIGHT TEST@'
015760	026463	051055	040505			
015766	042504	020122	044514			

```

015774 044107 020124 042524
016002 052123 100
016005 045 052045 051125 IM13A: .ASCII '%TURN LIGHT OF READER ON VIA RDR CONTROL SWITCH@'
016012 020116 044514 044107
016020 020124 043117 051040
016026 040505 042504 020122
016034 047117 053040 040511
016042 051040 051104 041440
016050 047117 051124 046117
016056 051440 044527 041524
016064 040110
016066 050045 052125 051440 IM13B: .ASCII '%PUT SW8 TO 1 TO TURN LIGHT OFF.@'
016074 034127 052040 020117
016102 020061 047524 052040
016110 051125 020116 044514
016116 044107 020124 043117
016124 027106 100
001450 .END START

```


PORC	006330	1190	1198#
PORD	006344	1199	1201#
PORE	006350	1198	1203#
POSA	006370	1211	1217#
POSB	006422	1216	1224#
POTA	006440	1231	1236#
POTLST=	177777	117#	1299
POTO	005266	894	902#
POT1	005316	903	917#
POT10	005654	1029	1047#
POT11	005714	1048	1065#
POT12	006016	1066	1096#
POT13	006060	1097	1115#
POT14	006130	1116	1135#
POT15	006204	1136	1159#
POT16	006262	1160	1182#
POT17	006354	1183	1208#
POT2	005346	918	932#
POT20	006424	1209	1228#
POT21	006452	1229	1242#
POT22	006510	1243	1259#
POT23	006562	1260	1278#
POT24	006640	1279	1298#
POT3	005422	933	954#
POT4	005456	955	970#
POT5	005510	971	988#
POT6	005542	989	1005#
POT7	005612	1006	1028#
POJA	006474	1245	1252#
POVA	006532	1262	1268#
POVB	006542	1270#	1273
POXA	006612	1281	1289#
POXB	006620	1291#	1294
POYA	006670	1301	1310#
POYB	006676	1312#	1315
P1AA	006756	1331	1336#
P1AB	006764	1335	1338#
P1BA	007006	1346	1351#
P1BB	007014	1350	1353#
P1CA	007030	1361	1365#
P1CB	007052	1367	1370#
P1DA	007104	1381	1385#
P1EA	007140	1397	1402#
P1FA	007162	1412	1416#
P1GA	007220	1429	1433#
P1HA	007266	1448	1455#
P1HB	007316	1454	1462#
P1IA	007330	1469	1473#
P1I3	007366	1474	1482#
P1IC	007402	1483	1485#
P1ID	007406	1482	1487#
P1JA	007426	1495	1502#
P1JB	007462	1501	1509#
P1KA	007502	1519	1526#
P1KB	007542	1525	1536#
P1LA	007560	1543	1549#

