```
                        .REM_
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZPCA-E-D

PRODUCT NAME: PC11 READER AND PUNCH TESTS

PROGRAM DATE: APRIL 1976

MAINTAINER: DIAGNOSTIC GROUP

```
                                1.      ABSTRACT
                                        --------

                                        THE PC11 READER AND PUNCH TESTS CONSISTS OF A PACKAGE OF TEST
                                        PROGRAMS DESIGNED TO TEST THE READER LOGIC, READER, PUNCH LOGIC,
                                        PUNCH, AND THE READER AND PUNCH IN COMBINATION, ALL TESTS ARE
                                        INCLUDED IN ONE OBJECT TAPE.

                                        THE AVAILABLE TESTS ARE LISTED HERE IN NUMERICAL ORDER:

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

                                        PROGRAMS PRG0 THROUGH PRG5 ARE THE READER AND PUNCH TESTS.
                                        PROGRAMS PRG6 THROUGH PRG13 ARE UTILITY ROUTINES THAT
                                        PRODUCE TEST TAPES AND AID IN MAKING ADJUSTMENTS.

                                2.      REQUIREMENTS
                                        ------------

                                2.1     EQUIPMENT
                                        ---------

                                        A.  PDP-11 SYSTEM, (8K MEMORY)
                                        B.  CONSOLE TELETYPE
                                        C.  PC11 READER OR PC11 READER AND PUNCH.

                                        THE PROCESSOR AND TELETYPE MUST BE IN OPERATING CONDITION.

                                        THE TELETYPE, HIGH SPEED READER, AND HIGH SPEED PUNCH MUST
                                        HAVE STANDARD PERIPHERAL ADDRESSES, REFER TO SECTION 7.3
                                        IF YOUR SYSTEM DOES NOT HAVE STANDARD PERIPHERAL ADDRESSES.

                                2.2     STORAGE
                                        -------

                                        THIS PROGRAM RUNS IN 8K MEMORY.

                                2.3     LOADING PRODEDURE
                                        -----------------

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

```
                        3.      SOFTWARE SWITCH REGISTER MANIPULATION
                                -------------------------------------


                        THIS PROGRAM DOES NOT MAKE USE OF THE HARDWARE SWITCH REGISTER
                        (LOC 177570). IT INSTEAD USES A SOFTWARE SWITCH REGISTER (SWREG)
                        LOCATED AT MEMORY ADDRESS 176. UPON EXECUTION OF EACH
                        SUB-PROGRAM WHICH ALLOWS SWREG SETTINGS, THE CONTENTS OS SWREG
                        ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND REQUESTS A NEW VALUE (IE)

                                SWR=XXXXXX      NEW=

                        POSSIBLE RESPONSES ARE:

                        1.      <CR>    IF NO CHANGES ARE TO BE MADE

                        2.    6 DIGITS  TO REPRESENT IN OCTAL THE NEW SWREG CONTENTS
                              0-7       LAST DIGIT FOLLOWED BY <CR>.

                        3.      ^U      TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED
                                        KEYING IN SWREG VALUE.

                        BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE
                        THE CONTENTS OF SWREG DURING PROGRAM EXECUTION.  BY TYPING A
                        ^G (CNTL G) ON THE CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG
                        TO CHANGE THE CONTENTS OF SWREG, WHICH WILL BE PROCESSED IN KEY AREAS
                        OF THE PROGRAM CODE, (IE) ERROR ROUTINES,AFTER COMMON HALTS
                        AND END OF PASS.

                        THE OPERATOR ALSO HAS THE ABILITY TO TYPE ^G OR ^U WHEN
                        INPUTTING DATA SUCH AS TEST NUMBER,ROUTINE NUMBER,AND ANY
                        ASCII DATA.

                        A.      ^G      WILL IMMEDIATELY EXECUTE THE ROUTINE TO CHANGE
                                        SWREG AND THEN RE-ASK QUESTION IN WHICH
                                        ^G WAS ANSWERED.

                        B.      ^U      WILL ALLOW OPERATOR TO REENTER DATA IF ERROR WAS COMMITTED.
```

```
                                4.      USE PROCEDURE
                                        -------------

                                        LOAD STARTING ADDRESS-PRESS START, THE PROGRAM IDENTIFIES ITSELF
                                        (1ST TIME THRU ONLY) AND REQUESTS THE PROGRAM NUMBER
                                        TO EXECUTE.
                                        THE FOLLOWING PAGES EXPLAIN IN DETAIL THE STEPS NECESSARY
                                        TO RUN EACH PROGRAM.

                                4.1     PRG0 USE PROCEDURE (DESCRIPTION IN SECTION 8.1
                                        ----------------------------------------------

                                        A.  INSURE THAT TELETYPE IS ON-LINE
                                        B.  HAVE AVAILABLE A TAPE LOOP OF SPECIAL BINARY COUNT PATTERN.
                                        C.  THE PROGRAM IDENTIFIES ITSELF AND TYPES OUT INSTRUCTIONS TO
                                            SELRCT ANY DESIRED SWREG OPTIONS

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

                                        BIT15=1  HALT ON ERROR.
                                        BIT14=1  ENTER SCOPE MODE.
                                        BIT13=1  INHIBIT ERROR PRINT.
                                        BIT11=1  INHIBIT ITERATION.
                                        BIT10=1  HALT AT END OF CURRENT TEST.
                                        BIT9=1   SELECT A SPECIFIC ROUTINE FOR EXECUTION.
                                        BIT8=1  BYPASS MANUAL INTERVENTION ROUTINES.

                                        D.  IF BIT9=1 THE PROGRAM REQUESTS THE SPECIFIC ROUTINE NUMBER.
                                        E.  REFER TO SECTION 6.2 IF ANY ERROR PRINTOUTS OCCUR.
                                        F.  THE PROGRAM RINGS THE BELL AT THE END OF EACH PASS.

                                        EXECUTION TIME.

                                        PRG0 IS USER DEPENDENT DUE TO THE USE OF MANUAL INTERVENTION
                                        ROUTINES. HOWEVER, WITH SWREG BITS SET TO  BYPASS MANUAL ROUTINES,
                                        ONE ERROR-FREE PASS WILL TAKE APPROXIMATELY 3 MINUTES.
```

```
                        4.2     PRG1 USE PROCEDURE (DESCRIPTION IN SECTION 8.2)
                                ----------------------------------------------

                                A.  INSURE THAT TELETYPE IS ON-LINE
                                B.  LOAD READER WITH SPECIAL BINARY COUNT PATTERN TEST TAPE LOOP.
                                    A TEST LOOP MUST BE USED, AS A NORMAL LENGTH TEST TAPE IS
                                    NOT LONG ENOUGH TO CONDUCT THE TEST. IF A TAPE LOOP IS NOT USED,
                                    DATA MUST BE POSITIONED OVER THE READ CELLS, NOT THE BLANK LEADER.
                                C.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO
                                    SET ANY DESIRED SWREG OPTIONS.

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

                                BIT15=1  HALT ON ERROR.
                                BIT14=1  ENTER SCOPE MODE.
                                BIT13=1  INHIBIT ERROR PRINT.
                                BIT11=1  INHIBIT ITERATION.
                                BIT10=1  HALT AT END OF CURRENT TEST.
                                BIT9=1   SELECT A SPECIFIC ROUTINE FOR EXECUTION.

                                D.  IF BIT9=1 THE PROGRAM REQUESTS A SPECIFIC ROUTINE NUMBER
                                E.  REFER TO SECTION 6.2 IF ANY ERROR PRINTOUTS OCCUR.
                                F.  THE PROGRAM RINGS THE BELL AT THE END OF EACH PASS.

                                EXECUTION TIME: ONE ERROR FREE PASS ABOUT 7 MINUTES.


                        4.3     PRG2 USE PROCEDURE (DESCRIPTION IN SECTION 8.3)
                                ----------------------------------------------

                                A.  INSURE THAT TELETYPE IS ON-LINE.
                                B.  INSURE THAT THE PUNCH HAS AN ADEQUATE SUPPLY OF TAPE.
                                C.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO
                                    SET ANY DESIRED SWREG OPTIONS.

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

                                BIT15=1  HALT ON ERROR.
                                BIT14=1  ENTER SCOPE MODE.
                                BIT13=1  INHIBIT ERROR PRINT.
                                BIT11=1  INHIBIT ITERATION.
                                BIT10=1 HALT AT END OF CURRENT TEST.
                                BIT9=1   SELECT A SPECIFIC ROUTINE FOR EXECUTION.
                                BIT8=1  BYPASS MANUAL INTERVENTION ROUTINES.

                                D.  IF BIT9=1 THE PROGRAM REQUESTS A SPECIFIC ROUTINE NUMBER.
                                E.  THE PROGRAM RINGS THE BELL AT THE END OF EACH PASS.
                                F.  REFER TO SECTION 6. IF ANY ERRORS OCCUR.

                                EXECUTION TIME

                                PRG2 IS USER DEPENDENT DUE TO THE USE OF MANUAL INTERVENTION
                                ROUTINES. WITH SWREG BIT8 SET TO BYPASS MANUAL ROUTINES, ONE
                                ERROR-FREE PASS WILL TAKE APPROXIMATELY 1.5 MINUTES.
```

```
                                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.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO
                                            SET ANY DESIRED SWREG OPTIONS.

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

                                        BIT13=1  INHIBIT ERROR PRINT.
                                        BIT11=1  INHIBIT ITERATION.
                                        BIT10=1 HALT AT END OF CURRENT TEST.
                                        BIT9=1   SELECT A SPECIFIC ROUTINE FOR EXECUTION.

                                        D.  IF BIT9=1 PROGRAM REQUESTS SPECIFIC ROUTINE NUMBER
                                        E.  UPON COMPLETION OF A PROGRAM PASS THE PROGRAM WILL TYPE
                                            "P0003 END" AND HALT. TO REPEAT PRESS CONTINUE.

                                        EXECUTION TIME: ONE PASS ABOUT 8 MINUTES.


                                4.5     PRG4 USE PROCEDURE (DESCRIPTION IN SECTION 8.5)
                                        ----------------------------------------------

                                        A.  INSURE THAT TELETYPE IS ON-LINE.
                                        B.  LOAD TAPE THAT WAS PUNCHED BY PRG3-PUNCH TEST IN READER.
                                            LOAD TAPE SO THAT THE FIRST RUBOUT  (ALL 1'S) IS 3 INCHES RIGHT
                                            OF THE METAL PLATE OVER THE READ STATION. MAKE READER READY.
                                        C.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO LOAD THE
                                            READER AND SELECT SWREG OPTIONS.
                                        D.   THE PROGRAM WILL READ THE TAPE AND REPORT
                                             ANY ERRORS. DISREGARD ANY ERRORS THAT OCCUR WHEN
                                            THE READER REACHES THE END OF THE TAPE.
                                        E.  THE SWREG OPTIONS FOR THIS PROGRAM ARE;

                                                BIT15=1  HALT ON ERROR.
                                                BIT13=1  INHIBIT ERROR PRINT.

                                        F.  REFER TO SECTION 6, IF ERRORS OCCUR.

                                        PRG4 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.
```

```
                                4.6     PRG5 USE PROCEDURE (DESCRIPTION IN SECTION 8.6)
                                        ----------------------------------------------

                                        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.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO PUNCH
                                            LEADER AND LOAD READER.
                                        E.   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.
                                        F.  THE SWREG OPTIONS AVAILABLE WITH THIS PROGRAM ARE:

                                                BIT15=1  HALT ON ERROR.
                                                BIT13=1  INHIBIT ERROR PRINT.

                                        I.  REFER TO SECTION 6, IF ERRORS OCCUR.

                                        EXECUTION TIME:  PRG5 IS CONTINUOUS RUNNING.

                                4.7     PRG6 USE PROCEDURE (DESCRIPTION IN SECTION 8.7)
                                        ----------------------------------------------

                                        THIS PROGRAM CONTINUOUSLY PUNCHES TAPE WITH 2 CHARACTERS SELECTED

                                        A.  INSURE THAT TELETYPE IS ON-LINE.
                                        B.  INSURE TAHT THE PUNCH HAS AN ADEQUATE SUPPLY OF TAPE.
                                        C.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO
                                            ENTER THE DESIRED ASCII CODES FOR CHARACTERS TO PUNCH.
                                        D.  PRESS CONTINUE, THE PROGRAM WILL PUNCH THE DESIRED
                                            CHARACTERS CONTINUOUSLY UNTIL STOPPED BY USER.

                                        EXECUTION TIME:  CONTINUOUS RUNNING PROGRAM.
```

```
                                4.8     PRG7 USE PROCEDURE (DESCRIPTION IN SECTION 8.8)
                                        ***********************************************

                                        THIS PROGRAM READS AND CHECKS A TAPE PUNCHED WITH ANY 2 CHARACTERS

                                        A.  INSURE THAT TELETYPE IS ON-LINE.
                                        B.  LOAD TAPE TO BE READ IN READER, DATA MUST BE UNDER READ STATION.
                                        C.  FOLLOW PROGRAM INSTRUCTIONS.
                                        D.  THE PROGRAM WILL READ THE TAPE AND REPORT ANY ERRORS.
                                        E.  THE SWREG OPTIONS AVAILABLE WITH THIS PROGRAM ARE:

                                                BIT15=1  HALT ON ERROR.
                                                BIT13=1  INHIBIT ERROR PRINT.

                                        F.  REFER TO SECTION 6, IF ERRORS OCCUR.

                                        EXECUTION TIME:  CONTINUOUS RUNNING PROGRAM.

                                4.9     PRG10 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.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO ENTER THE
                                            NUMBER OF CHARACTERS TO READ AND THE NUMBER OF MILLISECONDS
                                            TO STALL AFTER READING THE CHARACTERS. PLEASE NOTE:

                                                1. THE RANGE FOR CHARACTERS TO READ IS 1 THRU 377 (8).

                                                2. THE STALL VALUE MUST BE NON-ZERO,BETWEEN 1 AND 377(8).

                                        D.  PRESS CONTINUE. THE PROGRAM WILL CONTINUOUSLY READ AND STALL
                                            UNTIL STOPPED BY USER.

                                        EXECUTION TIME:  CONTINUOUS RUNNING PROGRAM.
```

```
                                4.10    PRG11 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.  THE PROGRAM IDENTIFIES ITSELF, AND TYPES INSTRUCTION
                                            TO MAKE THE PUNCH READY.
                                        D.  PRESS CONTINUE. THE SPECIAL BINARY COUNT PATTERN WILL
                                            BE PUNCHED UNTIL THE PROGRAM IS STOPPED BY USER.

                                4.11    PRG12 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.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO LOAD READER
                                            AND MAKE READY, AND TO SELECT DESIRED MEASUREMENT PERIOD.
                                        D.  PRESS CONTINUE WHEN READY TO START MEASUREMENT. THE READER
                                            WILL START RUNNING.
                                        E.  AT END OF TIME PERIOD,STRIKE ANY TTY KEY
                                            THE PROGRAM WILL TYPE AOUT THE READER SPEED IN CHARACTERS
                                            PER SECOND AND HALT.
                                        F.  TO REPEAT,PRESS CONTINUE WHEN READY
```

```
                        4.12    PRG13 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.  THE PROGRAM IDENTIFIES ITSELF AND TYPES INSTRUCTIONS TO MAKE
                                    PUNCH READY.
                                D.  PRESS CONTINUE WHEN READY TO START MEASUREMENT. THE PUNCH WILL
                                    START RUNNING.
                                E.  AT END OF TIME PERIOD (60 SECONDS),STRIKE ANY TTY KEY
                                    THE PROGRAM WILL TYPE OUT THE PUNCH SPEED IN CHARACTER PER
                                    SECOND AND HALT.
                                F.  TO REPEAT,PRESS CONTINUE
                                    CONTINUE WHEN READY.


                        5.      PROGRAM AND/OR OPERATOR ACTION
                                ------------------------------


                        5.1     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.

                                "INCORRECT PROGRAM SELECTED."
                                ----------------------------

                                THE USER HAS SELECTED FOR EXECUTION A NON-EXISTENT PROGRAM.
                                PRESS CONTINUE TO RETRY.

                                "INCORRECT ROUTINE SELECTED."
                                ----------------------------

                                THE USER HAS SELECTED FOR EXECUTION A NON-EXISTENT ROUTINE.
                                PRESS CONTINUE TO RETRY.
```

```
                                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 000176 - SYSTEM SOFTWARE AND INTERRUPT VECTOR AREA,
                                                                EXCEPT FOR PC11 AND TTY VECTORS.
```

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:

"ERROR P00XX T00YY PC 0ZZZZZ"

WHERE:
P00XX IS THE NUMBER OF THE PROGRAM BEING RUN,
T00YY IS THE NUMBER OF ROUTINE WHERE FAILURE OCCURRED.

PC 0ZZZZZ IS THE ADDRESS FROM WHICH THE ERROR CALLED WAS ISSUED.

WHEN THIS TYPE OF ERROR PRINTOUT OCCURS:

A. IN THE PROGRAM LISTING, LOOK UP THE ADDRESS REFERENCED BY PC0ZZZZZ,
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, AND ALSO IN THE "PROGRAM DESCRIPTION" SECTION OF THIS DOCUMENT.

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 "ERR1" SUBROUTINE WHICH IS CALLED BY AN "ERROR1" STATEMENT IN THE PROGRAM LISTING, A TYPICAL PRINTOUT WOULD LOOK AS FOLLOWS:

"ERROR P0005 T0000 PC 011350 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 ERR1 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 TEST17, 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 PRG4, 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 PRG4 WHEN IN READING THE LEADER THAT PRECEDES THE SPECIAL BINARY COUNT PUNCHED BY PRG3 THE DATA DOES NOT AGREE WITH THE EXPECTED DATA, CHECK THAT THE TAPE IS PUNCHED CORRECTLY, REFER TO PRG3 AND PRG4 DESCRIPTION.

"MATCH ERROR"

THIS PRINTOUT IS GENERATED BY PRG7 WHEN UNSUCCESSFUL IN MATCHING UP THE DATA READ FROM THE READER WITH THE EXPECTED DATA AS SPECIFIED, CHECK THAT THE TAPE IS THE ONE TO BE READ AND RESTAR" 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 PAT-
                                TERN 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     SWREG OPTIONS
                                        ----------

                                THE STANDARD SWREG OPTIONS ARE DESCRIBED HERE.

                                BIT15 = HALT ON ERROR.

                                BIT14 = 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.

                                BIT13 = INHIBIT ERROR PRINT.  THIS OPTION IF SET WILL REMOVE ALL
                                        ERROR PRINTOUTS.
```

(7,2 CONT'D)

BIT11 - 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 SWREG BIT11 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:

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.

BIT10 - HALT AT END OF CURRENT ROUTINE, FOR THOSE PROGRAMS CONSISTING OF A SET OF SEPARATE TEST ROUTINES, SWREG BIT10 SET TO A 1 CAUSES THE PROGRAM TO HALT AT THE COMPLETION OF THE ROUTINE CURRENTLY BEING EXECUTED, THREE POSSIBLE USES OF THIS OPTION ARE:

A. TO STEP THROUGH A PROGRAM ONE ROUTINE AT A TIME.
B. WHEN AN UNPREDICTED FAILURE HAS OCCURRED (BLOW UP, HANG UP) TO ADVANCE THROUGH THE PROGRAM ONE ROUTINE AT A TIME UNTIL THE BLOW UP OCCURS, THE ROUTINE FOLLOWING THE LAST IDENTIFIED ROUTINE WOULD BE THE FAILING ROUTINE.
C. WHEN A PROGRAM IS IN EXECUTION, TO DETERMINE HOW FAR THE PROGRAM HAS PROGRESSED.

(7,2 CONT'D)

BIT9 - SELECT ROUTINE, THE PROGRAMS THAT CONSIST OF INDIVIDUAL TEST ROUTINES, THE USER MAY ELECT TO RUN ONLY A SPECIFIED ROUTINE, TO SELECT A ROUTINE BIT 9 (SWREG) MUST BE SET THE PROGRAM THEN REQUESTS THE ROUTINE NUMBER TO BE RUN 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.

BIT8 - BYPASS MANUAL INTERVENTION ROUTINE, SOME PROGRAMS TEST ROUTINES REQUIRE THAT THE USER PERFORM SOME MANUAL OPERATION FOR WHICH THE PROGRAM HAS TO WAIT, THE USER MAY ELECT TO BYPASS THESE ROUTINES BY SETTING BIT8 OF SWREG, A GOOD POINT AT WHICH TO USE THIS OPTION WOULD BE AFTER A COMPLETE PASS HAS BEEN COMPLETED, AND THE USER WISHES TO LOOP THE PROGRAM WITHOUT HAVING TO INTERVENE, SELECTING A MANUAL ROUTINE WITH BIT9 OPTION AND BIT8 SET WILL CAUSE THE FOLLOWING PRINTOUT:

"?MANUAL ROUTINE, BIT8 IS SET."

EITHER TURN OFF BIT8, OR SELECT ANOTHER ROUTINE, PRESS CONTINUE.

7.3 TESTING PC11 AT NON-STANDARD ADDRESSES AND/OR VECTORS

THIS PROGRAM CAN TEST PC11'S ASSIGNED TO NON-STANDARD ADDRESSES AND VECTORS PROVIDED THOSE ADDRESSES AND VECTORS ARE PROVIDED TO THE PROGRAM AS FOLLOWS:

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

LOCATION	FROM STANDARD	TO NON-STANDARD
001210	177550	READER CSR ADDRESS
001212	177552	READER BUFFER ADDRESS
001214	177554	PUNCH CSR ADDRESS
001216	177556	PUNCH BUFFER ADDRESS
001220	000070	READER INTERRUPT VECTOR ADDRESS
001222	000200	READER PRIORITY LEVEL
001224	000074	PUNCH INTERRUPT VECTOR ADDRESS
001226	000200	PUNCH PRIORITY LEVEL.

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

LOCATION	FROM STANDARD	TO NON-STANDARD
001230	177560	TTY KEYBOARB CSR
001232	177562	TTY KEYBOARD BUFFER
001234	177564	TTY PRINTER CSR ADDRESS
001236	177566	TTY PRINTER BUFFER ADDRESS

```
                                    0.      DESCRIPTION
                                            -----------

                                    0.1     PRG0 PROGRAM DESCRIPTION
                                            ------------ -----------

                                            PRG0 TESTS THE PC11 INPUT LOGIC. THE PROGRAM CONSISTS OF 26
                                            TEST ROUTINES NUMBERED FROM 00 TO 30(8).

                                            RTN0 - TESTS THAT THE READER STATUS WORD (PRS) CAN BE
                                                   REFERENCED WITHOUT TRAPPING.

                                            RTN1 - TESTS THAT THE READER BUFFER (PRB) CAN BE REFERENCED
                                                   WITHOUT TRAPPING.

                                            RTN2 - MANUAL INTERVENTION ROUTINE, CHECKS THAT WITH PC11 POWER
                                                   OFF AND AFTER ISSUING A RESET THE ERROR BIT IS THE ONLY
                                                   BIT SET IN THE READER STATUS WORD (PRS).

                                            RTN3 - MANUAL INTERVENTION ROUTINE, CHECKS THAT THE ERROR BIT
                                                   (BIT 15) BECOMES SET IN PRS WITH READER OFF-LINE.

                                            RTN4 - MANUAL INTERVENTION ROUTINE, CHECKS THAT THE ERROR BIT
                                                   (BIT 15) BECOMES SET IN PRS WITH READER OUT-OF-TAPE.

                                            RTN5 - MANUAL INTERVENTION ROUTINE, CHECKS THAT THE ERROR BIT
                                                   (BIT 15) IS NOT SET (CLEARED) IN PRS WITH PC11 POWER ON,
                                                   READER ON-LINE, AND TAPE LOADED IN READER.

                                            RTN6 - TESTS ABILITY TO SET AND CLEAR THE INTERRUPT ENABLE
                                                   BIT IN PRS (BIT 6).

                                            RTN7 - TESTS ABILITY TO CLEAR THE INTERRRUPT ENABLE BIT IN
                                                   PRS (BIT 6) WITH A RESET INSTRUCTION.

                                           RTN10 - ENABLES READER, AND AFTER APPROXIMATELY 100 MILLISECONDS
                                                   CHECKS THAT THE DONE BIT HAS BECOME SET IN PRS (BIT 7).

                                           RTN11 - TESTS ABILITY TO READ THE DONE BIT RELIABLY (BIT 7 OF PRS).

                                           RTN12 - TESTS THAT RESET COMMAND CLEARS DONE BIT (BIT 7 OF PRS).

                                           RTN13 - TESTS THAT DONE BIT (BIT 7 OF PRS) IS CLEARED BY READER ENABLE.

                                           RTN14 - TESTS THAT DONE BIT (BIT 7 OF PRS) IS CLEARED BY REFERENCING
                                                   THE READER BUFFER (PRB).

                                           RTN15 - TEST THAT ENABLING READER (BIT 0 OF PRS) SETS THE BUSY
                                                   BIT (BIT 11 OF PRS).
```

```
                                (8.1 CONT'D)

        RTN16 - TESTS ABILITY TO READ THE BUSY BIT RELIABLY (BIT 11 OF PRS).

        RTN17 - TESTS ABILITY TO READ THE READER BUFFER (PRB) RELIABLY.

        RTN20 - TESTS THAT THE READER BUFFER (PRB) IS CLEARED BY
                READER ENABLE.

        RTN21 - TESTS THAT READER INTERRUPTS ON DONE. IF THE INTERRUPT
                IS SERVICED, IT INDICATES THAT THE READER IS INTERRUPTING
                AT THE CORRECT VECTOR ADDRESS.

        RTN22 - TESTS THAT THE READER DOES NOT INTERRUPT WITH PROCESSOR
                SET TO THE SAME PRIORITY AS THE READER.

        RTN23 - TESTS THAT THE READER INTERRUPTS WITH PROCESSOR SET
                TO A PRIORITY ONE LEVEL LOWER THAN THE READER'S.

        RTN24 - CHECKS THAT THE READER DOES NOT REINTERRUPT AFTER
                AN RTI COMMAND WHEN THE DONE BIT IS LEFT SET.

        RTN25 - CHECKS THAT THE READER INTERRUPTS IMMEDIATELY UPON LOWERING
                CP PRIORITY TO 0.

        RTN26 - MANUAL INTERVENTION ROUTINE, CHECKS THAT ERROR BIT SET
                (BIT 15 OF PRS) CRIPPLES READER ENABLE.

        RTN27 - MANUAL INTERVENTION ROUTINE, CHECKS THAT THE ERROR BIT
                IS ABLE TO INTERRUPT, AND DOES NOT REINTERRUPT AFTER SERVICE.

        RTN30 - MANUAL INTERVENTION ROUTINE, CHECKS THAT AFTER AN ERROR
                INTERRUPT HAS BEEN SERVICED ISSUING A READER ENABLE
                CAUSES AN IMMEDIATE INTERRUPT.
```

```
                                0.2     PRG1 PROGRAM DESCRIPTION
                                        ------------------------

                                        PRG1 IS THE PC11 READER DATA TEST. IT CONSISTS OF 3 ROUTINES
                                        NUMBERED FROM 00 TO 02. THE PROGRAM USES A SPECIAL BINARY COUNT
                                        PATTERN TEST TAPE LOOP IN ALL ROUTINES.

                                        RTN0 - READS AND CHECKS 10000 CHARACTERS AT FULL SPEED.

                                        RTN1 - READS AND CHECKS 500 CHARACTERS. A STALL OF BETWEEN
                                               0 AND 7 MILLISECONDS OCCURS BETWEEN EACH CHARACTER.

                                        RTN2 - READS AND CHECKS 1000 GROUPS OF 3 CHARACTERS EACH.
                                               A STALL OF BETWEEN 0 TO 31 MSECS OCCURS BETWEEN
                                               EACH CHARACTER GROUP.

                                        RTN3 - READS AND CHECKS 1000 GROUPS OF CHARACTERS.  CHARACTER
                                               LENGTH VARIES RANDOMLY BETWEEN 1 AND 15.  A STALL OF
                                               BETWEEN 0 TO 31 MSECS OCCURS BETWEEN EACH CHARACTER GROUP.

                                        RTN4 - READS AND CHECKS 1000 GROUPS OF CHARACTERS, THE NUMBER
                                               OF CHARACTERS IN A GROUP VARIES RANDOMLY BETWEEN 1 AND 77.
                                               A STALL OF BETWEEN 0 TO 31 MSECS OCCURS BETWEEN EACH
                                               GROUP OF CHARACTERS.

                                        IN ALL ROUTINES, THE PROGRAM WILL AUTOMATICALLY RESYNC ITSELF
                                        TO THE TEST TAPE AFTER THREE ERRORS HAVE OCCURRED.

                                0.3     PRG2 PROGRAM DESCRIPTION
                                        ------------------------

                                        PRG2 TESTS THE PC11 OUTPUT LOGIC, THE PROGRAM CONSISTS OF 17
                                        TEST ROUTINES NUMBERED FROM 00 TO 20 (8).

                                        RTN0 - TESTS THAT THE PUNCH STATUS WORD (PPS) CAN BE REFERENCED
                                               WITHOUT TRAPPING.

                                        RTN1 - TESTS THAT THE PUNCH BUFFER (PPB) CAN BE REFERENCED WITHOUT
                                               TRAPPING.

                                        RTN2 - MANUAL INTERVENTION ROUTINE, CHECKS THAT WITH PC11 POWER
                                               OFF AND AFTER ISSUING A RESET, THE ERROR AND READY BITS
                                               ARE THE ONLY BITS SET IN THE PUNCH STATUS WORD (PPS).

                                        RTN3 - MANUAL INTERVENTION ROUTINE, CHECKS THAT THE ERROR BIT
                                               (BIT 15 OF PPS) BECOMES SET WHEN THE PUNCH IS OUT OF TAPE.
```

(8,3 CONT'D)

RTN4 - MANUAL INTERVENTION ROUTINE, CHECKS THAT THE ERROR BIT DOES NOT SET WITH PC11 POWER ON, AND TAPE IN PUNCH.

RTN5 - TESTS ABILITY TO SET AND CLEAR THE INTERRUPT ENABLE BIT (BIT 6 IN PPS).

RTN6 - TESTS ABILITY TO CLEAR THE INTERRUPT ENABLE BIT WITH RESET INSTRUCTION.

RTN7 - TESTS THAT THE READY BIT (BIT 7 OF PPS) IS SET BY A RESET INSTRUCTION, AND THAT THE BIT CAN BE READ RELIABLY.

RTN10 - TESTS THAT THE READY BIT (BIT 7 OF PPS) IS CLEARED BY LOADING THE PUNCH BUFFER (PPB).

RTN11 - TESTS THAT THE READY BIT (BIT 7 OF PPS) IS NOT CLEARED BY BYTE LOADING PPB+1.

RTN12 - TESTS THAT THE READY BIT (BIT 7 OF PPS) IS ABLE TO INTERRUPT. IF THAT INTERRUPT IS SERVICED, IT INDICATES THAT INTERRUPT IS OCCURRING AT THE CORRECT VECTOR ADDRESS.

RTN13 - TESTS THAT THE READY BIT DOES NOT REINTERRUPT AFTER IT HAS BEEN SERVICED AND THE READY BIT LEFT ON.

RTN14 - TESTS THAT THE PUNCH DOES NOT INTERRUPT WITH THE PROCESSOR AT SAME PRIORITY LEVEL AS THE PUNCH.

RTN15 - TESTS THAT THE PUNCH INTERRUPTS WITH PROCESSOR SET TO A PRIORITY ONE LEVEL LOWER THAN THE PUNCH'S.

RTN16 - TESTS THAT THE PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING OF PROCESSOR PRIORITY TO LEVEL 0.

RTN17 - TEST THAT THE PUNCH ERROR BIT (BIT 15 OF PPS) IS ABLE TO INTERRUPT, AND THAT IT DOES NOT REINTERRUPT AFTER BEING SERVICED.

RTN20 - MANUAL INTERVENTION ROUTINE, CHECKS THAT AFTER AN ERROR INTERRUPT HAS BEEN SERVICED, LOADING THE PUNCH BUFFER CAUSES AN IMMEDIATE INTERRUPT.

```
                                0.4     PRG3 PROGRAM DESCRIPTION
                                        ------------------------

                                        PRG3 EXERCISES THE PUNCH, THE PROGRAM CONSISTS OF 4 ROUTINES
                                        NUMBERED FROM 00 TO 03, THE DATA USED FOR OUTPUT IS THE SPECIAL
                                        BINARY COUNT PATTERN, ALL ROUTINES PUNCH DATA BLOCKS IN THE
                                        FOLLOWING FORMAT:

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

                                        RTN0 - PUNCHES 5 DATA BLOCKS AT FULL SPEED,

                                        RTN1 - PUNCHES 5 DATA BLOCKS, THE SPECIAL BINARY COUNT PATTERN
                                               DATA IS PUNCHED WITH RANDOM STALLS OF UP TO 47 MILLISECONDS
                                               AFTER EACH CHARACTER,

                                        RTN2 - 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,

                                        RTN3 - PUNCHES 1 DATA BLOCK, THE SPECIAL BINARY COUNT PATTERN
                                               DATA IS PUNCHED WITH A 5 SECOND STALL PRECEDING EACH
                                               32 CHARACTER GROUP PUNCHED,

                                0.5     PRG4 PROGRAM DESCRIPTION
                                        ------------------------

                                        PRG4 VERIFIES THE PAPER TAPE PRODUCED BY PRG3, 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 ROUTINE #, 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,
```

```
                                8.6     PRG5 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,

                                8.7     PRG6 PROGRAM DESCRIPTION
                                        ------------------------

                                        PRG6 WILL PUNCH CONTINUOUSLY THE 2 CHARACTERS WHOSE ASCII CODES HAVE
                                        BEEN SELECTED, THE ROUTINE IS USED FOR GENERATING ALL 0'S TAPE,
                                        ALL 1'S TAPE, ONES AND ZEROES TAPE, ETC,

                                8.8     PRG7 PROGRAM DESCRIPTION
                                        ------------------------

                                        PRG7 READS AND CHECKS A TAPE PUNCHED WITH THE CHARACTERS WHOSE
                                        ASCII CODES HAVE BEEN SELECTED, THIS ROUTINE IS USEFUL IN SETTING UP
                                        THE READ PHOTOCELLS AND READ AMPLIFIER,

                                8.9     PRG10 PROGRAM DESCRIPTION
                                        -------------------------

                                        PRG10 WILL ENABLE THE READER FOR THE NUMBER OF
                                        CHARACTERS SPECIFIED, AND THEN IT WILL STALL
                                        FOR THE NUMBER OF MILLISECONDS SPECIFIED,
                                        THIS ROUTINE IS USEFUL IN SETTING UP THE READER CLOCK,
                                        ACCELERATOR, STROBE, AND FOR CHECKING THE STOP DELAY,
```

```
1017
1018
1019                                    8.10    PRG11 PROGRAM DESCRIPTION
1020                                            ------------------------
1021
1022                                            PRG11 PUNCHES THE SPECIAL BINARY COUNT PATTERN CONTINUOUSLY.
1023
1024                                    8.11    PRG12 PROGRAM DESCRIPTION
1025                                            ------------------------
1026
1027                                            PRG12 IS A ROUTINE USED TO CHECK THE SPEED OF THE READER.
1028                                            READER SPEED CAN BE MEASURED IN TWO WAYS:
1029
1030                                            A.  COARSE, 30 SECOND TIMING, PLUS OR MINUS 10 CHARACTER ACCURACY.
1031                                            B.  FINE,  300 SECOND TIMING, PLUS OR MINUS 1 CHARACTER ACCURACY.
1032
1033
1034                                            THE USER CONTROLS THE DURATION OF THE TIMING PERIOD BY USING A
1035                                            SWEEP SECOND HAND WATCH OR STOP-WATCH. AT THE END OF THE
1036                                            TIMING PERIOD, STRIKE ANY TTY KEY TO OBTAIN A SPEED PRINTOUT.
1037
1038                                    8.12    PRG13 PROGRAM DESCRIPTION
1039                                            ------------------------
1040
1041                                            PRG13 IS USED TO CHECK THE SPEED OF THE PUNCH. THE ROUTINE
1042                                            USES A 60 SECOND TIMING PERIOD THAT IS CONTROLLED BY THE USER.
1043                                            AT THE END OF THE TIMING PERIOD STRIKE ANY TTY KEY TO OBTAIN A
1044                                            SPEED PRINTOUT.
```

```
                                        .ABS
                                        .TITLE  PC11 READER-PUNCH TESTS
                                        .NLIST  MC,MD,TOC
                                        .LIST   ME
                                ;PRG0 - READER LOGIC TESTS
                                ;PRG1 - READER TEST
                                ;PRG2 - PUNCH LOGIC TESTS
                                ;PRG3 - PUNCH TEST
                                ;PRG4 - PUNCH VERIFY ROUTINE
                                ;PRG5 - COMBINED READER-PUNCH TEST
                                ;PRG6 - PUNCH 2 CHARACTERS FROM SR,
                                ;PRG7 - READ 2 CHARACTERS AS PER SR,
                                ;PRG10 - READ X CHARS, STALL Y MSECS,
                                ;PRG11 - PUNCH SPECIAL BINARY COUNT PATTERN TAPE,
                                ;PRG12 - READER SPEED PRINT ROUTINE,
                                ;PRG13 - PUNCH SPEED PRINT ROUTINE,

                  000000                .=0
000000  000002                .+2                     ;UNASSIGNED TRAP
000002  000000                HALT
000004  000006        MACHER: .+2                     ;SP OVERFLOW, BUS ERROR TRAP
000006  000000                HALT
000010  000012                .+2                     ;RESERVED INSTRUCTION TRAP
000012  000000                HALT
000014  000016                .+2                     ;TRACE TRAP
000016  000000                HALT
000020  000022                .+2                     ;TRAP TO CALL IOX
000022  000000                HALT
000024  000026                .+2                     ;POWER FAIL TRAP
000026  000000                HALT
000030  002442                EMTINT                  ;EMT TRAP
000032  000340                PRTY7
000034  004310                DLYX
000036  000340                PRTY7
                        ;LOCATIONS 40 THROUGH 776 ARE FILLED WITH .+2 AND HALT,
                  000000        EMTX=0
                  177776        CC=177776
                  177776        PSW=177776              ;PS ADDRESS
                  001200        SPBOT=1200
                  000240        NOP=240
                  000000        OPEN=0                  ;SUBJECT TO PROGRAM MODIFICATION
                  100000        MANUAL=BIT15
                  000000        R0=%0
                  000006        R6=%6
                  000007        PC=%7
                  100000        BIT15=100000            ;BIT DEFINITIONS
                  040000        BIT14=40000
                  020000        BIT13=20000
                  010000        BIT12=10000
                  004000        BIT11=4000
                  002000        BIT10=2000
                  001000        BIT9=1000
                  000400        BIT8=400
                  000200        BIT7=200
```

```
1101            000100                  BIT6=100
1102            000040                  BIT5=40
1103            000020                  BIT4=20
1104            000010                  BIT3=10
1105            000004                  BIT2=4
1106            000002                  BIT1=2
1107            000000                  BIT0=0
1108            005726                  POPSP=5726              ;POP THE STACK, SAME AS TST (6)+
1109            022626                  POPSP2=022626           ;POP STACK TWICE, SAME AS CMP (6)+,(6)+
1110            000340                  PRTY7=340               ;PRIORITY LEVEL DEFINITIONS
1111            000300                  PRTY6=300
1112            000240                  PRTY5=240
1113            000200                  PRTY4=200
1114            000140                  PRTY3=140
1115            000100                  PRTY2=100
1116            000040                  PRTY1=40
1117            000000                  PRTY0=0
1118            104400                  DELAYX=TRAP+0
1119            000007                  BELL=007
1120
1121            000046                  .=46
1122  000046    002340                  LOGIC
1123
1124
1125                                    ;*****************************************************************
1126                                    ;NOTE:  PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT
1127                                            ;A HARDWARE SWITCH REGISTER-REFER TO DOCUMENT
1128                                    ;*****************************************************************
1129
1130            000174                  .=174
1131  000174    000000                DISPREG:OPEN
1132  000176    000000                SWREG:  OPEN
1133
1134
1135            000200                  .=200
1136  000200    000167  001226                JMP     START           ;GO TO START OF PROGRAM.
1137            001204                  .=.+1000
1138  001204    000176                SWR:    SWREG
1139  001206    000174                DISPLAY:DISPREG
1140  001210    177550                PRS:    177550                  ;READER CSR
1141  001212    177552                PRB:    177552                  ;READER BUFFER
1142  001214    177554                PPS:    177554                  ;PUNCH CSR
1143  001216    177556                PPB:    177556                  ;PUNCH BUFFER
1144  001220    000070                RDRVTR: 70                      ;READER INTERRUPT VECTOR
1145  001222    000200                RDRLVL: PRTY4                   ;READER PRIORITY LEVEL
1146  001224    000074                PCHVTR: 74                      ;PUNCH INTERRUPT VECTOR
1147  001226    000200                PCHLVL: PRTY4                   ;PUNCH PRIORITY LEVEL
1148  001230    177560                TKS:    177560                  ;LSR CSR
1149  001232    177562                TKB:    177562                  ;LSR BUFFER
1150  001234    177564                TPS:    177564                  ;LSP CSR
1151  001236    177566                TPB:    177566                  ;LSP BUFFER
1152  001240    000000                PRGNUM: OPEN                    ;CONTAINS CURRENT PROGRAM#
1153  001242    000000                BRCTR:  OPEN
1154  001244    000000                DVDND:  OPEN
1155  001246    000000                DVQUOT: OPEN
1156  001250    000000                MSEC:   OPEN
```

```
1157  001252  000000          KSTART: OPEN            ;CURRENT PROGRAM START ADDRESS.
1158  001254  000000          CURTST: OPEN            ;CONTAINS ADDR OF CURRENT TEST.
1159  001256  000000          RTNNO:  OPEN            ;CONTAINS CURRENT TEST #.
1160  001260  000000          NXTST:  OPEN            ;CONTAINS ADDR OF NEXT TEST.
1161  001262  000000          ICTR:   OPEN            ;CONTAINS CURRENT ITERATION COUNT
1162  001264  000000          SCOPTR: OPEN            ;CONTAINS CURRENT SCOPE POINTER.
1163  001266  000000          PRGID:  OPEN            ;CONTAINS PROGRAM INDICATORS
1164  001270  005376          PRGTAB: PRG0            ;PRG0 START ADDRESS
1165  001272  007666                  PRG1            ;PRG1 START ADDRESS
1166  001274  010216                  PRG2            ;PRG2 START ADDRESS
1167  001276  011634                  PRG3            ;PRG3 START ADDRESS
1168  001300  012302                  PRG4            ;PRG4 START ADDRESS
1169  001302  012660                  PRG5            ;PRG5 START ADDRESS
1170  001304  013454                  PRG6            ;PRG6 START ADDRESS
1171  001306  013606                  PRG7            ;PRG7 START ADDRESS
1172  001310  014134                  PRG10           ;PRG10 START ADDRESS
1173  001312  014322                  PRG11           ;PRG11 START ADDRESS
1174  001314  014372                  PRG12           ;PRG12 START ADDRESS
1175  001316  014524                  PRG13           ;PRG13 START ADDRESS
1176  001320                  EMTTAB:
1177  001320  003566                  DLY             ;POINTER FOR EMT CALL DELAY
1178  001322  002514                  EHLT            ;POINTER FOR EMT CALL EHALT
1179  001324  003012                  SRSETT          ;POINTER FOR EMT CALL SRESET
1180  001326  003400                  TYP             ;POINTER FOR EMT CALL TYPE
1181  001330  003534                  TYPS            ;POINTER FOR EMT CALL TYPES
1182  001332  004262                  STALL           ;POINTER FOR EMT CALL STALL
1183  001334  003234                  ERR             ;POINTER FOR EMT CALL ERROR
1184  001336  003244                  ERR1            ;POINTER FOR EMT CALL ERROR1
1185  001340  002462                  CHLT            ;POINTER FOR EMT CALL CHALT
1186  001342  002732                  STPTRV          ;POINTER FOR EMT CALL STRDRV
1187  001344  002762                  STPTPV          ;POINTER FOR EMT CALL STPCHV
1188  001346  002110                  CHAIN           ;POINTER FOR EMT CALL SCOPE
1189  001350  014722                  OPTS            ;POINTER FOR EMT CALL OPTSEL
1190  001352  015156                  CNTLU           ;POINTER FOR EMT CALL CNTL
1191  001354  014654                  TTIN            ;POINTER FOR EMT CALL TTYIN
1192  001356  014740                  VALINP          ;POINTER FOR EMT CALL VALID
1193  001360  015120                  CKSWRR          ;POINTER FOR EMT CALL CKSWR
1194
1195
1196  001362  000000          ERRT:   OPEN
1197  001364  000000          TMP1:   OPEN
1198  001366  000000          TMP2:   OPEN
1199  001370  000001          FRST:   1
1200  001372  000000          COUNT:  OPEN
1201  001374  000000          TIB:    OPEN
1202  001376  000000          RCNT:   OPEN            ;CHARACTER COUNT
1203  001400  000000          CRBUF:  OPEN            ;HOLDS ONE CHARACTER FROM READER.
1204  001402  000000          CHR1:   OPEN
1205  001404  000000          CHR2:   OPEN
1206  001406  000000          CHR3:   OPEN
1207  001410  000000          CHR1A:  OPEN
1208  001412  000000          CHR2A:  OPEN
1209  001414  000000          CHR3A:  OPEN
1210  001416  000000          EPCTR:  OPEN
1211  001420  000000          CTRA:   OPEN
1212  001422  000000          CTRB:   OPEN
```

```
1213  001424  000000                  CTRC:   OPEN
1214  001426  000000                  CTRD:   OPEN
1215  001430  000000                  XCNT:   OPEN
1216  001432  012706  001200          START:  MOV     #SPBOT,%6       ;SET BOTTOM OF SP STACK.
1217  001436  005067  176334                  CLR     PSW
1218  001442  005767  177722                  TST     FRST
1219  001446  001404                          BEQ     1$
1220  001450  104003                          TYPE
1221  001452  020773                          $TITLE
1222  001454  005067  177710                  CLR     FRST
1223  001460  013746  000004          1$:     MOV     @#4,-(R6)
1224  001464  012737  001662  000004        MOV     #XORA,@#4
1225  001472  012737  000433  177060        MOV     #433,@#177060
1226  001500  012637  000004                MOV     (%6)+,@#4
1227  001504  012737  177777  002040        MOV     #-1,@#XORFLG
1228
1229  001512  012767  000026  177530        MOV     #26,MSEC
1230  001520  104003                          TYPE
1231  001522  002042                          MESS
1232  001524  012767  160000  177456        MOV #160000,PRS         ;XOR PRS ADDRESS
1233  001532  012767  160002  177452        MOV #160002,PRB         ;XOR PRB ADDRESS
1234  001540  012767  160004  177446        MOV #160004,PPS         ;XOR PPS ADDRESS
1235  001546  012767  160006  177442        MOV #160006,PPB         ;XOR PPB ADDRESS
1236  001554  012767  000770  177436        MOV #770,RDRVTR         ;XOR READER VECTOR
1237  001562  012767  000774  177434        MOV #774,PCHVTR         ;XOR PUNCH VECTOR
1238  001570  012767  000006  176206  INGXOR: MOV #6,MACHER
1239  001576  005067  177454                  CLR     RTNNO
1240  001602  012767  000003  177562  1$:     MOV     #3,COUNT
1241  001610  012767  020314  013106        MOV     #$$TEST,TLX
1242  001616  104014                          OPTSEL
1243  001620  022767  000003  177544        CMP     #3,COUNT
1244  001626  001765                          BEQ     1$
1245  001630  016700  177530                  MOV     TMP1,%0
1246  001634  005067  177570                  CLR     XCNT            ;INIT THE XOR PROGRAM CONTROL
1247  001640  042700  177760                  BIC     #177760,%0      ;LIMIT (SR) TO BITS 3-0
1248  001644  020027  000013                  CMP     %0,#13          ;COMPARE (SR) TO PROGRAM LIMIT
1249  001650  101410                          BLOS    CRTA            ;VALID PROGRAM NUMBER?
1250  001652  104003                          TYPE                    ;TYPE INCORRECT PROGRAM MESSAGE.
1251  001654  015267                          CM2
1252  001656  104010                          CHALT                   ;COMMON HALT.
1253  001660  000664                          BR      START           ;START OVER.
1254  001662  022626          XORA:   CMP     (R6)+,(R6)+
1255  001664  012637  000004                MOV     (R6)+,@#4
1256  001670  000737                          BR      INGXOR
1257  001672  005067  177370          CRTA:   CLR     PRGID
1258
1259
1260
1261  001676  010067  177336                  MOV     %0,PRGNUM       ;SAVE PROGRAM NUMBER AT PRGNUM
1262  001702  006300          CRTB:   ASL     %0              ;R0X2
1263  001704  000170  001270                  JMP     @PRGTAB(0)      ;GO TO SELECTED PROGRAM.
1264  001710  016767  177336  177342  GETRDY: MOV     KSTART,NXTST    ;ADDR OF 1ST ROUTINE TO NXTST
1265  001716  012767  000006  176060  CLEAN:  MOV     #6,MACHER       ;SET UP BUS ERRROR TRAP.
1266  001724  012706  001200                MOV     #SPBOT,R6       ;SET UP STACK.
1267  001730  104002                          SRESET
1268  001732  005067  176040                  CLR     PSW
```

```
1269  001736  004767  000422          GTRDYA: JSR     %7,FORWD        ;ROLL FORWARD TO "NEXT" ROUTINE.
1270  001742  032777  001000  177234  GTRDYB: BIT     #BIT9,@SWR      ;SELECT ROUTINE?
1271  001750  001003                          BNE     GTRDYC          ;BR IF YES.
1272  001752  004767  000440                  JSR     %7,GOTST        ;GO RUN ROUTINE.
1273  001756  000532                          BR      CHNB            ;NO GO, MANUAL RTN BYPASSED.
1274  001760  012767  000003  177404  GTRDYC: MOV     #3,COUNT
1275  001766  012767  020246  012730          MOV     #SRTN,TLX
1276  001774  104014                          OPTSEL
1277  001776  022767  000003  177366          CMP     #3,COUNT
1278  002004  001765                          BEQ     GTRDYC
1279  002006  016700  177352                  MOV     TMP1,%0
1280  002012  042700  177600                  BIC     #177600,%0      ;MASK UNDESIRED BITS
1281  002016  126700  177234  NTYET:  CMPB    RTNNO,%0        ;COMPARE RTNNO TO (R0)
1282  002022  001017                          BNE     GTRDYD          ;BRANCH IF ROUTINE NOT FOUND YET.
1283  002024  004767  000366                  JSR     %7,GOTST        ;GO RUN ROUTINE.
1284  002030  104003                          TYPE                    ;NO GO, MANUAL RTN BYPASSED.
1285  002032  015467                          CM5                     ;TYPE MESSAGE.
1286  002034  104010                          CHALT
1287  002036  000724                          BR      GETRDY
1288  002040  000000          XORFLG: 0
1289  002042  021445  041520  030461  MESS:   .ASCII  '%#PC11 XOR TST#'
1290  002050  054040  051117  052040
1291  002056  052123     100
1292            002062                        .EVEN
1293  002062  022767  177777  177170  GTRDYD: CMP     #-1,NXTST       ;NO, CHECK FOR LAST ROUTINE.
1294  002070  001403                          BEQ     INCRTN
1295  002072  004767  000266                  JSR     %7,FORWD
1296  002076  000747                          BR      NTYET
1297  002100  104003          INCRTN: TYPE                    ;TYPE INCORRECT RTN MESSAGE.
1298  002102  015325                          CM3
1299  002104  104010                          CHALT                   ;COMMON HALT.
1300  002106  000700                          BR      GETRDY          ;START OVER.
1301  002110  012706  001200          CHAIN:  MOV     #SPBOT,R6       ;RESTORE STACK.
1302  002114  104020                          CKSWR
1303  002116  005737  002040                  TST     @#XORFLG        ;IS XOR TESTER HERE?
1304  002122  100011                          BPL     3$              ;BR IF NOT
1305  002124  013746  000004                  MOV     @#4,-(%6)       ;SAVE MEM. 4
1306  002130  012737  002354  000004          MOV     #XOR,@#4
1307  002136  005737  177060                  TST     @#177060        ;IS XOR IN ERROR?
1308  002142  012637  000004                  MOV     (%6)+,@#4       ;NO,REPLACE MEM. 4
1309  002146  032777  040000  177030  3$:     BIT     #BIT14, @SWR    ;SCOPE?
1310  002154  001404                          BEQ     S2$             ;BR IF NOT
1311  002156  005067  175614          S1$:    CLR     PSW
1312  002162  000177  177076                  JMP     @SCOPTR         ;GO TO  SCOPE ENTRY
1313  002166  032777  004000  177010  S2$:    BIT     #BIT11,@SWR     ;INHIBIT ITERATION?
1314  002174  001003                          BNE     CHNAA           ;BR IF YES.
1315  002176  005367  177060                  DEC     ICTR            ;NO, ICTR 0?
1316  002202  001365                          BNE     S1$             ;BR IF NOT
1317  002204  032777  002000  176772  CHNAA:  BIT     #BIT10,@SWR     ;HALT AT END OF TEST?
1318  002212  001414                          BEQ     CHNB            ;BR IF NOT.
1319  002214  005067  177144                  CLR     TMP1
1320  002220  116767  177032  177136          MOVB    RTNNO,TMP1
1321  002226  004567  002622                  JSR     %5,ACNV4
1322  002232  001364                          TMP1
1323  002234  020756                          RTNN
1324  002236  104003                          TYPE
```

```
1325  002240  020736                          ENDRTN
1326  002242  104010                          CHALT
1327  002244  032777  001000  176732  CHNB:   BIT     #BIT9,@#WR      ;SELECT ROUTINE?
1328  002252  001216                          BNE     GETRDY          ;BR IF YES.
1329  002254  022767  177777  176776          CMP     #-1,NXTST       ;NO. LAST TEST?
1330  002262  001215                          BNE     CLEAN           ;BR IF NOT.
1331  002264  005767  177550                  TST     XORFLG
1332  002270  100015                          BPL     1$
1333  002272  005167  177132                  COM     XCNT
1334  002276  005767  177126                  TST     XCNT
1335  002302  100005                          BPL     2$
1336  002304  012767  010240  176740          MOV     #CT0,KSTART     ;START PUN LOGIC TESTS IF XOR
1337  002312  000167  177372                  JMP     GETRDY
1338  002316  012767  005432  176726  2$:     MOV     #AT0,KSTART     ;START RDR LOGIC TESTS IF XOR
1339  002324  104003                  1$:     TYPE                    ;TYPE PROGRAM END MESSAGE.
1340  002326  015263                          APGEND
1341  002330  013700  000042                  MOV     @#42,R0         ;GET CONTENTS OF 42.
1342  002334  001405                          BEQ     HERE            ;BR IF 0.
1343  002336  000005                          RESET
1344  002340  004710                  LOGIC:  JSR     PC,(0)          ;RETURN TO MONITOR.
1345  002342  000240  000240  000240          .WORD   NOP,NOP,NOP
1346  002350  000167  177334          HERE:   JMP     GETRDY          ;REPEAT.
1347  002354  022626                  XOR:    CMP     (SP)+,(SP)+     ;POP STACK
1348  002356  012637  000004                  MOV     (SP)+,@#4       ;REPLACE MEM 4
1349  002362  000675                          BR      $18             ;GO TO SCOPE ENTRY
1350  002364  016705  176670          FORWD:  MOV     NXTST,R5        ;ADDR OF NEXT ROUTINE TO R5.
1351  002370  012567  176662                  MOV     (R5)+,RTNNO     ;GET NEXT ROUTINE NUMBER.
1352  002374  012567  176660                  MOV     (R5)+,NXTST     ;GET ADDR OF NEXT "NEXT" ROUTINE.
1353  002400  012567  176656                  MOV     (R5)+,ICTR      ;GET ITERATION COUNT.
1354  002404  012567  176654                  MOV     (R5)+,SCOPTR    ;GET SCOPE LOOP ENTRY POINTER.
1355  002410  010567  176640                  MOV     R5,CURTST       ;ADDR OF NOW CURRENT TEST TO CURTST.
1356  002414  000207                          RTS     R7              ;EXIT FORWD SUBROUTINE.
1357  002416  005767  176634          GOTST:  TST     RTNNO   ;CHECK FOR MANUAL RTN.
1358  002422  100005                          BPL     GOTSTA          ;BRANCH IF NOT MANUAL RTN.
1359  002424  032777  000400  176552          BIT     @#BIT8,@#WR     ;MANUAL RTN. BYPASS IT?
1360  002432  001401                          BEQ     GOTSTA          ;NO. RUN IT.
1361  002434  000207                          RTS     R7              ;BYPASS MANUAL ROUTINE.
1362  002436  000177  176612          GOTSTA: JMP     @CURTST         ;GO RUN TEST
1363                                  ;EMT INTERPRETER ROUTINE.
1364  002442  010046                  EMTINT: MOV     R0,-(SP)        ;PUSH R0.
1365  002444  016600  000002                  MOV     2(SP),R0        ;GET EMT PC.
1366  002450  014000                          MOV     -(R0),R0        ;GET EMT CALL.
1367  002452  006300                          ASL     R0              ;TIMES 2.
1368  002454  016000  171320                  MOV     EMTTAB-10000(R0),R0 ;DEVELOP EMT RTN ADDR.
1369  002460  000200                          RTS     R0              ;GO TO EMT RTN. RESTORE R0.
1370                                  ;COMMON HALT ROUTINE.
1371  002462  011600                  CHLT:   MOV     (SP),R0
1372  002464  005740                          TST     -(R0)
1373  002466  010067  176672                  MOV     R0,TMP1
1374  002472  004567  002330                  JSR     R5,ACNV6
1375  002476  001364                          TMP1
1376  002500  020717                          GWAS
1377  002502  104003                          TYPE
1378  002504  020711                          PCHLT
1379  002506  000000                          HALT
1380  002510  104020                          CKSWR
```

```
1381  002512  000002                          RTI
1382                                  ;ERROR HALT ROUTINE.
1383  002514  005777  176464          EHLT:   TST     @#WR            ;CHECK FOR HALT ON ERROR.
1384  002520  100002                          BPL     EHLTA           ;BRANCH IF NO HALT DESIRED.
1385  002522  000000                          HALT
1386  002524  104020                          CKSWR
1387  002526  000002                  EHLTA:  RTI                     ;EXIT
1388                                  ;ROUTINE TO CHECK FOR READER ERROR.
1389  002530  005777  176454          ARDER:  TST     @PRS            ;TEST ERROR BIT IN PRS
1390  002534  100401                          BMI     1$              ;BRANCH IF ERROR BIT SET.
1391  002536  000207                          RTS     %7              ;NOT SET, EXIT.
1392  002540  104004                  1$:     TYPES                   ;TYPE STATUS MESSAGE AND
1393  002542  017404                          BMI                     ;INSTRUCTIONS
1394  002544  016334                          IM6
1395  002546  177777                          -1
1396  002550  104010                          CHALT                   ;HALT TO WAIT FOR USER.
1397  002552  000766                          BR      ARDER           ;GO TEST AGAIN.
1398                                  ;DD11-XOR PROGRAMMABLE SIMULATOR OF PCOS (PUNCH/READER)
1399                                          ;CALL   -JSR    %5,PCSIN
1400                                          ;       SIMULATOR CONSTANT
1401                                          ;       @LABLE OF NEXT INSTRUCTION IF ON XOR TESTER
1402                                          ;       IF NOT ON AN XOR,THIS ROUTINE EXIT TO THE INSTRUCTION FOLLOWING THE CALL
1403  002554  005767  177260          PCSIN:  TST     XORFLG          ;ARE WE ON AN XOR TESTER
1404  002560  001425                          BEQ     RETRN           ;IF NOT ON AN XOR TESTER RETURN
1405  002562  013746  000004                  MOV     @4,-(%6)        ;SAVE TRAP CATCHER
1406  002566  012737  002630  000004          MOV     #1$,@4          ;IF XOR TRAPS DURRING LOAD GO TO 1$
1407
1408
1409
1410
1411  002574  052777  000001  176406          BIS     #1,@PRS         ;YES,INHIBIT A M SIGNAL FROM CAUSING ERROR DUE DIFFERENT
1412  002602  104000                          DELAY                   ;CIRCUIT DELAYS AT THE TEST HEAD
1413  002604  000001                          1
1414  002606  012537  177060          2$:     MOV     (%5)+,@#177060  ;LOAD SIMULATOR
1415
1416
1417  002612  104000                          DELAY                   ;WAIT FOR ERROR BIT TO SETTLE
1418  002614  000050                          5#
1419
1420  002616  012637  000004          3$:     MOV     (%6)+,@4        ;REPLACE TRAP CATCHER
1421  002622  000005                          RESET
1422  002624  011505                          MOV     (%5),%5         ;RETURN TO TEST SETUP
1423  002626  000205                          RTS     %5              ;RETURN TO TEST
1424  002630  022626                  1$:     CMP     (%6)+,(%6)+     ;FIX STACK
1425  002632  000771                          BR      3$              ;CONTINUE WITH  THE SIM ROUTINE
1426  002634  062705  000004          RETRN:  ADD     #4,%5           ;NOT AN XOR TESTER ,RETURN TO PROGRAM AFTER PCSIM CALL
1427  002640  000205                          RTS     %5
1428
1429                                  ;ROUTINE TO CHECK FOR READER READY.
1430  002642  032777  004000  176340  ARRDY:  BIT     #4000,@PRS      ;TEST BUSY BIT.
1431  002650  001001                          BNE     ARRDYA          ;BRANCH IF BUSY BIT SET.
1432  002652  000207                          RTS     %7              ;READER READY, EXIT.
1433  002654  004767  000036          ARRDYA: JSR     %7,TSM2         ;TYPE STATUS AND INSTRUCTION MESSAGE.
1434  002660  000770                          BR      ARRDY           ;GO CHECK AGAIN
1435                                  ;ROUTINE TO FETCH A CHARACTER
1436  002662  004767  177754          AREAD:  JSR     %7,ARRDY        ;CHECK FOR READER READY.
```

```
 1437  002666  105277  176316          AREAD1: INCB    @PRS            ;ENABLE READER
 1438  002672  005777  176312          ARDA:   TST     @PRS            ;TEST ERROR BIT
 1439  002676  100404                          B<I     ARDB            ;BRANCH IF ERROR BIT SET.
 1440  002700  105777  176304                  TSTB    @PRS            ;CHECK DONE BIT
 1441  002704  100372                          BPL     ARDA            ;BRANCH IF NOT DONE.
 1442  002706  000207                          RTS     %7              ;DONE, EXIT.
 1443  002710  004767  000002          ARDB:   JSR     %7,TSM2         ;TYPE STATUS AND INSTRUCTION MESSAGE.
 1444  002714  000762                          BR      AREAD           ;TRY AGAIN.
 1445  002716  104004                  TSM2:   TYPES                   ;TYPE READER NOT READY STATUS
 1446  002720  017433                          SM2                     ;MESSAGE AND HALT.
 1447  002722  016334                          IM6
 1448  002724  177777                          -1
 1449  002726  104010                          CHALT
 1450  002730  000207                          RTS     %7              ;EXIT
 1451                                  ;ROUTINE TO SET READER INTERRUPT VECTOR AND PRIORITY
 1452  002732  017667  000000  000012  STPTRV: MOV     @(6),STPRA+2    ;MOVE VECTOR ADDR TO STPRA+2
 1453  002740  062716  000002                  ADD     #2,@#6          ;SET UP EXIT
 1454  002744  016701  176250                  MOV     RDRVTR,%1
 1455  002750  012721  000000          STPRA:  MOV     #OPEN,(1)+      ;SET VECTOR ADDRESS
 1456  002754  016721  176242                  MOV     RDRLVL,(1)+     ;SET PRIORITY
 1457  002760  000002                          RTI                     ;EXIT
 1458                                  ;ROUTINE TO SET PUNCH INTERRUPT VECTOR AND PRIORITY.
 1459  002762  017667  000000  000012  STPTPV: MOV     @(6),STPPA+2    ;MOVE VECTOR ADDR TO STPPA+2
 1460  002770  062716  000002                  ADD     #2,@#6          ;SET UP EXIT
 1461  002774  016701  176224                  MOV     PCHVTR,%1
 1462  003000  012721  000000          STPPA:  MOV     #OPEN,(1)+      ;SET VECTOR ADDRESS.
 1463  003004  016721  176216                  MOV     PCHLVL,(1)+     ;SET PRIORITY
 1464  003010  000002                          RTI                     ;EXIT.
 1465                                  ;ROUTINE TO ISSUE RESET.
 1466  003012  012700  052525          SRSETT: MOV     #52525,%0               ;DATA TO R0.
 1467  003016  005100                          COM     %0              ;COMPLEMENT (R0).
 1468  003020  010067  177770                  MOV     %0,SRSETT+2     ;(R0) TO SRSETT+2.
 1469  003024  000005                          RESET                   ;ISSUE RESET. (R0) IS
 1470  003026  000002                          RTI                     ;DISPLAYED, EXIT.
 1471                                  ;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
 1472  003030  016700  000042          RNGEN:  MOV     RP1,%0
 1473  003034  006100                          ROL     %0
 1474  003036  006100                          ROL     %0
 1475  003040  066700  000034                  ADD     RP2,%0
 1476  003044  010067  000026                  MOV     %0,RP1
 1477  003050  006100                          ROL     %0
 1478  003052  006100                          ROL     %0
 1479  003054  066700  000020                  ADD     RP2,%0
 1480  003060  006100                          ROL     %0
 1481  003062  006100                          ROL     %0
 1482  003064  010067  000010                  MOV     %0,RP2
 1483  003070  016700  000002                  MOV     RP1,%0
 1484  003074  000207                          RTS     %7              ;EXIT, NUMBER IN R0
 1485  003076  001233                  RP1:    1233
 1486  003100  007622                  RP2:    7622
 1487                                  ;SUBROUTINE TO READ CHARACTER FROM READER USING INTERRUPT.
 1488  003102  104011                  BREAD:  STRDRV                  ;SET READER VECTOR
 1489  003104  003162                          BREADB                  ;TO BREADB
 1490  003106  012767  000340  174662          MOV     #PRTY7,PSW      ;SET PRIORITY 7.
 1491  003114  004767  177522                  JSR     %7,ARRDY        ;CHECK FOR READER READY.
 1492  003120  052777  000101  176062          BIS     #101,@PRS       ;ENABLE PTR AND PTRI.
```

```
1493  003126  104000                        DELAY                   ;WAIT FOR READER INTERRUPT.
1494  003130  000226                        150.
1495  003132  005077  176052                CLR     @PRS            ;CLEAR PTRI ENABLE.
1496  003136  104093                        TYPE                    ;TYPE NO PTR RESPONSE
1497  003140  020124                        EM7                     ;MESSAGE
1498  003142  000757                        BR      BREAD           ;TRY AGAIN.
1499  003144  017767  176042  176226  BREADA: MOV     @PRB,CRBUF      ;CHAR READ TO CRBUF.
1500  003152  022626                        POP&P2
1501  003154  005067  174616                CLR     PSW             ;CLEAR STATUS.
1502  003160  000207                        RTS     %7              ;EXIT SUBROUTINE.
1503  003162  005077  176022          BREADB: CLR     @PRS            ;CLEAR PTR INTERRUPT ENABLE.
1504  003166  005777  176016                TST     @PRS            ;TEST FOR ERROR.
1505  003172  100411                        BMI     BREADC          ;BRANCH IF ERROR.
1506  003174  105777  176010                TSTB    @PRS            ;TEST FOR DONE BIT.
1507  003200  100403                        BMI     BRDBB           ;BRANCH IF DONE BIT SET.
1508  003202  104007                        ERROR1                  ;ERROR,FALSE READER INTERRUPT.
1509  003204  020140                        EM1@
1510  003206  000405                        BR      &RDCC
1511  003210  012716  003144          BRDBB: MOV     #BREADA,@%6     ;MODIFY INTERRUPT EXIT TO BREADA.
1512  003214  000002                        RTI                     ;OK, EXIT INTERRUPT.
1513  003216  004767  177474          BREADC: JSR     %7,TSM2         ;TYPE NOT READY MESSAGE.
1514  003222  012716  003230          BRDCC:  MOV     #BRDDD,@%6      ;SET UP TO RETRY.
1515  003226  000002                        RTI                     ;EXIT INTERRUPT.
1516  003230  022626                  BRDDD:  POP&P2
1517  003232  000723                        BR      BREAD           ;GO TRY AGAIN.
1518                                  ;ERROR ROUTINES
1519  003234  004767  000060          ERR:    JSR     %7,ERRA         ;FIRST ERROR TYPEOUT
1520  003240  104001                        EHALT                   ;GO HALT IF HALT SWITCH IS SET.
1521  003242  000002                        RTI                     ;EXIT.
1522  003244  004767  000050          ERR1:   JSR     %7,ERRA         ;FIRST ERROR TYPEOUT
1523  003250  004767  000024                JSR     %7,INHPRT       ;INHIBIT PRINT?
1524  003254  000406                        BR      ERR1A           ;NO PRINT
1525  003256  011600                        MOV     @%6,%@          ;DEVELOP ADDRESS OF ADDITIONAL
1526  003260  011067  000002                MOV     @%@,.+6         ;ERROR TYPEOUT.
1527  003264  104003                        TYPE                    ;ADDITIONAL ERROR TYPEOUT.
1528  003266  000000                        OPEN
1529  003270  104001                        EHALT                   ;GO HALT IF HALT SWITCH IS SET.
1530  003272  062716  000002          ERR1A:  ADD     #2,@%6          ;SET UP EXIT.
1531  003276  000002                        RTI                     ;EXIT
1532  003300  104020                  INHPRT: CKSWR
1533  003302  032777  020000  175674        BIT     #BIT13,@SWR     ;INHIBIT PRINT?
1534  003310  001002                        BNE     .+6             ;BR IT YES.
1535  003312  062716  000002                ADD     #2,@%6          ;NO.
1536  003316  000207                        RTS     %7              ;EXIT.
1537  003320  016600  000002          ERRA:   MOV     2(6),R@         ;GET EMT PC.
1538  003324  005740                        TST     -(@)            ;DECREMENT BY 2.
1539  003326  010067  176030                MOV     %@,ERRT         ;ADDRESS OF ERROR CALL TO ERRT
1540  003332  004767  177742                JSR     %7,INHPRT       ;INHIBIT PRINT?
1541  003336  000207                        RTS     %7              ;NO PRINT.
1542  003340  004567  001462                JSP     %5,ACNV6        ;CONVERT ERROR CALL ADDRESS TO ASCII.
1543  003344  001362                        ERRT
1544  003346  017623                        APC
1545  003350  004567  001500                JSR     %5,ACNV4        ;CONVERT PROGRAM@ TO ASCII
1546  003354  001240                        PRGNUM
1547  003356  017602                        APNUMB
1548  003360  004567  001470                JSR     %5,ACNV4        ;CONVERT TEST@ TO ASCII
```

```
1549  003364  001256                          RTNNO
1550  003366  017612                          ATNUMB
1551  003370  104003                          TYPE                    ;TYPE ERROR MESSAGE
1552  003372  017570                          EMB
1553  003374  104020                          CKSWR
1554  003376  000207                          RTS     %7              ;EXIT.
1555                                  ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
1556  003400  011600                  TYP:    MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
1557  003402  062716  000002                  ADD     #2,@%6          ;SET UP EXIT.
1558  003406  011000                          MOV     @%0,%0          ;ADDRESS OF MESSAGE TO R0.
1559  003410  112067  000116          TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
1560  003414  122767  000100  000110          CMPB    #100,TYPDAT     ;CHECK FOR"@"CHARACTER
1561  003422  001006                          BNE     TYPC            ;BRANCH IF NOT"@".
1562  003424  112767  000177  000100          MOVB    #177,TYPDAT     ;OUTPUT RUBOUT.
1563  003432  004767  000030                  JSR     %7,TYPD
1564  003436  000002                          RTI                     ;TERMINATOR CHAR. DONE. EXIT.
1565  003440  122767  000045  000064  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR"%".
1566  003446  001416                          BEQ     TYPF            ;BRANCH IF"%"
1567  003450  122767  000043  000054          CMPB    #43,TYPDAT      ;NOT"%",CHECK FOR"+".
1568  003456  001417                          BEQ     TYPG            ;BRANCH IF "+"
1569  003460  004767  000002                  JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
1570  003464  000751                          BR      TYPA
1571  003466  116777  000040  175542  TYPD:   MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
1572  003474  105777  175534                  TSTB    @TPS            ;WAIT FOR DONE FLAG.
1573  003500  100375                          BPL     .-4
1574  003502  000207                          RTS     %7              ;EXIT
1575  003504  112767  000015  000020  TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
1576  003512  004767  177750                  JSR     %7,TYPD         ;GO TYPE CHAR.
1577  003516  112767  000012  000006  TYPG:   MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT.
1578  003524  004767  177736                  JSR     %7,TYPD         ;GO TYPE CHAR.
1579  003530  000727                          BR      TYPA
1580  003532  000000                  TYPDAT: OPEN
1581                                  ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
1582  003534  011600                  TYPS:   MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
1583  003536  062716  000002                  ADD     #2,@%6          ;UPDATE TO NEXT MESSAGE ADDRESS
1584  003542  011067  000014                  MOV     @%0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
1585  003546  022767  177777  000006          CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
1586  003554  001001                          BNE     TYPSA           ;BRANCH IF NOT TERMINATOR.
1587  003556  000002                          RTI                     ;TERMINATOR, EXIT
1588  003560  104003                  TYPSA:  TYPE                    ;CALL ON TYP SUB TO TYPE MESSAGE
1589  003562  000000                  TYPSB:  OPEN                    ;ADDRESS OF MESSAGE GOES HERE
1590  003564  000763                          BR      TYPS            ;GO PROCESS NEXT MESSAGE
1591                                  ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
1592  003566  011667  000124          DLY:    MOV     @%6,DLCNT       ;GET ADDRESS THAT CONTAINS DELAY COUNT
1593  003572  062716  000002                  ADD     #2,@%6          ;SET UP EXIT
1594  003576  017767  000114  000112          MOV     @DLCNT,DLCNT    ;MILLISECONDS COUNT TO DLCNT
1595  003604  005067  174166                  CLR     PSW
1596  003610  016767  175434  000076  DLYA:   MOV     MSEC,DLCTR      ;MOVE 1 MSEC DELAY CONSTANT TO DLCTR
1597  003616  016767  000072  000070  DLYB:   MOV     DLCTR,DLCTR
1598  003624  016767  000064  000062          MOV     DLCTR,DLCTR
1599  003632  016767  000056  000054          MOV     DLCTR,DLCTR
1600  003640  016767  000050  000046          MOV     DLCTR,DLCTR
1601  003646  016767  000042  000040          MOV     DLCTR,DLCTR
1602  003654  016767  000034  000032          MOV     DLCTR,DLCTR
1603  003662  016767  000026  000024          MOV     DLCTR,DLCTR
1604  003670  016767  000020  000016          MOV     DLCTR,DLCTR
```

```
 1605  003676  005367  000012            DEC     DLCTR           ;DECREMENT 1 MSEC COUNTER
 1606  003702  001345                    BNE     DLYB            ;BRANCH IF NOT YET 1 MILLISECOND
 1607  003704  005367  000006            DEC     DLCNT           ;DECREMENT MSECS COUNT (DLCNT)
 1608  003710  001337                    BNE     DLYA            ;BRANCH IF DLCNT NOT 0
 1609  003712  000002                    RTI                     ;DONE DELAYING,EXIT
 1610  003714  000000            DLCTR:  OPEN                    ;1 MILLISECONDS COUNT
 1611  003716  000000            DLCNT:  OPEN                    ;CONTAINS MILLISECONDS COUNT
 1612                            ;ROUTINE TO CALIBRATE DELAY ROUTINE USING READER.
 1613          004136                    TMCON=RTINT8+2
 1614  003720  012700  000006    RTMCAL: MOV     #6,R0           ;SET UP TO READ 6 CHARS.
 1615  003724  012767  000021  000204    MOV     #17,,TMCON      ;TIME TO READ 6 CHARS TO TMCON.
 1616  003732  104011                    STRDRV                  ;SET READER VECTOR.
 1617  003734  004044                    RTMINT
 1618  003736  005067  175300            CLR     BRCTR
 1619  003742  012777  000101  175240    MOV     #101,@PRS       ;ENABLE READER AND INTERRUPTS.
 1620  003750  005067  174022    RTMCLA: CLR     PSW
 1621  003754  016767  175262  175260 RTMCLB: MOV     BRCTR,BRCTR
 1622  003762  016767  175254  175252    MOV     BRCTR,BRCTR
 1623  003770  016767  175246  175244    MOV     BRCTR,BRCTR
 1624  003776  016767  175240  175236    MOV     BRCTR,BRCTR
 1625  004004  016767  175232  175230    MOV     BRCTR,BRCTR
 1626  004012  016767  175224  175222    MOV     BRCTR,BRCTR
 1627  004020  016767  175216  175214    MOV     BRCTR,BRCTR
 1628  004026  016767  175210  175206    MOV     BRCTR,BRCTR
 1629  004034  005267  175202            INC     BRCTR
 1630  004040  001345                    BNE     RTMCLB          ;BR IF RESULT NOT 0.
 1631  004042  104010                    CHALT                   ;BRCTR OVERFLOWED.
 1632
 1633  004044  005777  175140    RTMINT: TST     @PRS            ;READER ERROR?
 1634  004050  100405                    BMI     RTMERR          ;BR IF YES.
 1635  004052  005300                    DEC     R0              ;READ 6 CHARS?
 1636  004054  001420                    BEQ     RTINTA          ;BR IF YES.
 1637  004056  005277  175126            INC     @PRS            ;NO. ENABLE READER.
 1638  004062  000002                    RTI                     ;EXIT INTERRUPT.
 1639  004064  004767  176626    RTMERR: JSR     PC,TSM2         ;READER ERROR.
 1640  004070  012716  003720            MOV     #RTMCAL,(6)     ;GO TRY AGAIN.
 1641  004074  000002                    RTI
 1642  004076  104004    PTMERR: TYPES                           ;PUNCH ERROR.
 1643  004100  017456                    $M3
 1644  004102  016620                    IM16
 1645  004104  177777                    -1
 1646  004106  104010                    CHALT
 1647  004110  012716  004170            MOV     #PTMCAL,(6)     ;GO TRY AGAIN.
 1648  004114  000002                    RTI
 1649  004116  005077  175066    RTINTA: CLR     @PRS            ;DISABLE READER INTERRUPTS.
 1650  004122  005067  175120            CLR     DVQUOT          ;CLEAR QUOTIENT.
 1651  004126  016767  175110  175110    MOV     BRCTR,DVDND
 1652  004134  162767  000000  175102 RTINTB: SUB     #0,DVDND        ;DIVIDE DVDND BY 17 OR 100
 1653  004142  103403                    BLO     RTINTC
 1654  004144  005267  175076            INC     DVQUOT          ;+1 TO QUOTIENT.
 1655  004150  000771                    BR      RTINTB          ;REPEAT SUBTRACTION.
 1656  004152  016767  175070  175070 RTINTC: MOV     DVQUOT,MSEC     ;1MSEC CONSTANT TO MSEC.
 1657  004160  005067  173612            CLR     PSW
 1658  004164  022626                    POPSP2
 1659  004166  000207                    RTS     PC              ;EXIT.
 1660                            ;ROUTINE TO CALIBRATE DELAY ROUTINE USING PUNCH.
```

I3

```
1661  004170  005000                PTMCAL: CLR     R0              ;GET PUNCH RUNNING.
1662  004172  004767  001040                JSR     PC,MSPCH
1663  004176  012700  000005                MOV     #5,R0           ;SET UP TO PUNCH 5 CHARS.
1664  004202  012767  000144  177726        MOV     #100.,TMCON     ;TIME TO PUNCH 5 CHARS TO TMCON.
1665  004210  104012                        STPCHV                  ;SET PUNCH INTERRUPT VECTOR.
1666  004212  004234                        PTMINT
1667  004214  005067  175022                CLR     BRCTR
1668  004220  005077  174772                CLR     @PPB            ;OUTPUT A 0.
1669  004224  052777  000100  174762        BIS     #BIT6,@PPS      ;ENABLE PUNCH INTERRUPTS.
1670  004232  000646                        BR      RTMCLA
1671  004234  005777  174754        PTMINT: TST     @PPS            ;PUNCH ERROR?
1672  004240  100716                        BMI     PTMERR          ;BR IF YES.
1673  004242  005300                        DEC     R0              ;PUNCHED 5 CHARS?
1674  004244  001403                        BEQ     PTINTA          ;BR IF YES.
1675  004246  005077  174744                CLR     @PPB            ;OUTPUT ANOTHER 0.
1676  004252  000002                        RTI                     ;EXIT INTERRUPT.
1677  004254  005077  174734        PTINTA: CLR     @PPS            ;DISABLE INTERRUPTS.
1678  004260  000720                        BR      RTINTA+4
1679                                ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS, MAXIMUM STALL
1680                                ;DETERMINED BY CONTENTS OF LOC STLMSK.
1681  004262  004767  176542        STAL:   JSR     %7,RNGEN        ;GO GET RANDOM NUMBER.
1682  004266  046700  000014                BIC     STLMSK,%0       ;0 IN R0, APPLY STALL MASK.
1683  004272  001404                        BEQ     STALB           ;BRANCH IF RESULT IS 0.
1684  004274  010067  000002                MOV     %0,STALA
1685  004300  104000                        DELAY                   ;DELAY
1686  004302  000000        STALA:  OPEN                    ;DELAY COUNT
1687  004304  000002        STALB:  RTI                     ;DONE, EXIT.
1688  004306  000000        STLMSK: OPEN                    ;STALL MASK.
1689                                ;SUB TO DELAY X TIME.
1690          004314                DLYX0=DLYX+4
1691          004326                DLYX1=DLYXA+4
1692  004310  012727  000040  000000 DLYX:   MOV     #40,%0          ;SET UP COUNT OF 40.
1693  004316  005067  173454                CLR     PSW
1694  004322  012727  001750  000000 DLYXA:  MOV     #1000.,%0       ;SET DELAY.
1695  004330  005367  177772        DLYXB:  DEC     DLYX1           ;DECREMENT DLYX1.
1696  004334  001375                        BNE     DLYXB           ;BR IF NOT 0 RESULT.
1697  004336  005367  177752                DEC     DLYX0           ;DECREMENT DLYX0.
1698  004342  001367                        BNE     DLYXA           ;BR IF NOT 0 RESULT.
1699  004344  000002                        RTI                     ;EXIT.
1700                                ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT (1-77)
1701  004346  004767  176456        GRCNT:  JSR     %7,RNGEN        ;GET RANDOM NUMBER
1702  004352  046700  000010                BIC     RCMSK,%0        ;APPLY MASK
1703  004356  001773                        BEQ     GRCNT           ;TRY AGAIN IF RESULT 0
1704  004360  010067  000004                MOV     %0,RNCNT        ;COUNT TO RNCNT
1705  004364  000207                        RTS     %7              ;EXIT.
1706  004366  000000        RCMSK:  OPEN                    ;RANDOM CHARACTER MASK.
1707  004370  000000        RNCNT:  OPEN                    ;RANDOM CHARACTER COUNT.
1708                                ;SUBROUTINE TO COMPARE DATA READ FROM READER AGAINST EXPECTED DATA AND REPORT ERRORS.
1709  004372  004767  000314        BCHECK: JSR     %7,GTBIN        ;GET BIN CHARACTER(IN R0)
1710  004376  020067  174776                CMP     %0,CRBUF        ;COMPARE(R0)TO DATA IN CRBUF
1711  004402  001001                        BNE     .+4             ;BRANCH IF NOT SAME(ERROR).
1712  004404  000207                        RTS     %7              ;OK,EXIT.
1713  004406  010067  174750                MOV     %0,ERRT
1714  004412  004567  000436                JSR     %5,ACNV4
1715  004416  001362                        ERPT
1716  004420  017655                        ASB
```

```
1717  004422  004567  000426                  JSR     %5,ACNV4
1718  004426  001400                          CRBUF
1719  004430  017670                          AWAS
1720  004432  104007                          ERROR1
1721  004434  017632                          EM1
1722  004436  005367  174754                  DEC     ERCTR           ;DECREMENT ERROR COUNTER
1723  004442  001002                          BNE     .+6             ;BRANCH IF NO THIRD ERROR
1724  004444  004767  000002                  JSR     %7,BSYNC        ;RESYNC THE READER.
1725  004450  000207                          RTS     %7              ;EXIT.
1726                                ;SUBROUTINE TO SYNC THE READER TO A SPECIAL BINARY COUNT PATTERN TEST TAPE.
1727  004452  004767  000176          BSYNC:  JSR     %7,INBIN        ;INITIALIZE BINARY PATTERN
1728  004456  004767  176420                  JSR     %7,BREAD        ;READ CHAR.
1729  004462  004767  176414                  JSR     %7,BREAD        ;READ CHAR.
1730  004466  004767  176410                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR1
1731  004472  016767  174702  174702          MOV     CRBUF,CHR1
1732  004500  004767  176376                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR2
1733  004504  016767  174670  174672          MOV     CRBUF,CHR2
1734  004512  004767  176364                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR3.
1735  004516  016767  174656  174662          MOV     CRBUF,CHR3
1736  004524  004767  000012                  JSR     %7,SYNCA        ;GO SYNC
1737  004530  000750                          BR      BSYNC           ;NO SYNC, TRY AGAIN.
1738  004532  012767  000003  174656          MOV     #3,ERCTR
1739  004540  000207                          RTS     %7              ;SUCCESS,EXIT.
1740  004542  012767  001000  000102  SYNCA:  MOV     #512.,SYCTRA    ;512 TO SYCTRA.
1741  004550  004767  000136  SYNCB:  JSR     %7,GTBIN        ;BIN CHAR TO CHR1A.
1742  004554  010067  174630                  MOV     %0,CHR1A
1743  004560  004767  000126                  JSR     %7,GTBIN        ;BIN CHAR TO CHR2A.
1744  004564  010067  174622                  MOV     %0,CHR2A
1745  004570  004767  000116                  JSR     %7,GTBIN        ;BIN CHAR TO CHR3A.
1746  004574  010067  174614                  MOV     %0,CHR3A
1747  004600  026767  174576  174602          CMP     CHR1,CHR1A      ;CHR1 AND CHR1A SAME?
1748  004606  001013                          BNE     SYNCC           ;BR IF NOT.
1749  004610  026767  174570  174574          CMP     CHR2,CHR2A      ;CHR2 AND CHR2A SAME?
1750  004616  001007                          BNE     SYNCC           ;BR IF NOT.
1751  004620  026767  174562  174566          CMP     CHR3,CHR3A      ;CHR3 AND CHR3A SAME?
1752  004626  001003                          BNE     SYNCC           ;BR IF NOT.
1753  004630  062716  000002                  ADD     #2,(6)          ;SET UP SYNCED EXIT.
1754  004634  000207                          RTS     %7              ;EXIT.
1755  004636  005367  000010          SYNCC:  DEC     SYCTRA          ;TRIED 512 TIMES?
1756  004642  001342                          BNE     SYNCB           ;BR IF NOT.
1757  004644  104007                          ERROR1                  ;SYNC ERROR MESSAGE.
1758  004646  017747                          EM3
1759  004650  000207                          RTS     %7              ;NO SYNC EXIT.
1760  004652  000000                  SYCTRA: OPEN
1761                                ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
1762  004654  012767  177777  000014  INBIN:  MOV     #-1,RIND        ;SET ALL VARIABLES
1763  004662  004567  000300                  JSR     %5,BMOVE        ;TO MINUS 1.
1764  004666  004676                          RIND
1765  004670  004677                          RIND+1
1766  004672  000013                          11.
1767  004674  000207                          RTS     %7              ;EXIT
1768  004676  000000                  RIND:   OPEN
1769  004700  000000                  PT0:    OPEN
1770  004702  000000                  PT1:    OPEN
1771  004704  000000                  PIND:   OPEN
1772  004706  000000                  PT0P:   OPEN
```

```
1773  004710  000000                    PT1P:   OPEN
1774                                    ;SPECIAL BINARY COUNT PATTERN SUBROUTINE, EXITS WITH BIN CHAR IN R0
1775  004712  016767  177762  177762    GTBIN:  MOV     PT0,PT1         ;PREVIOUS BIN CHAR TO PT1
1776  004720  005167  177756            COM     PT1
1777  004724  005167  177746            COM     RIND
1778  004730  001002                    BNE     .+6
1779  004732  005267  177744            INC     PT1
1780  004736  042767  177400  177736    BIC     #177400,PT1     ;MASK TO 8 BITS
1781  004744  016767  177732  177726    MOV     PT1,PT0         ;SAVE BIN CHAR IN PT0
1782  004752  016700  177724            MOV     PT1,%0          ;BIN CHAR TO R0.
1783  004756  000207                    RTS     %7              ;EXIT.
1784  004760  016767  177722  177722    GTBINP: MOV     PT0P,PT1P       ;PREVIOUS BIN CHAR TO PT1P
1785  004766  005167  177716            COM     PT1P
1786  004772  005167  177706            COM     PIND
1787  004776  001002                    BNE     .+6
1788  005000  005267  177704            INC     PT1P
1789  005004  042767  177400  177676    BIC     #177400,PT1P    ;MASK TO 8 BITS.
1790  005012  016767  177672  177666    MOV     PT1P,PT0P       ;SAVE BIN CHAR IN PT0P.
1791  005020  016701  177664            MOV     PT1P,%1         ;BIN CHAR TO R1.
1792  005024  000207                    RTS     %7              ;EXIT.
1793                                    ;OCTAL TO ASCII CONVERT ROUTINES
1794  005026  012500                    ACNV6:  MOV     (5)+,%0         ;CONVERT TO 6 ASCII, GET OCTAL ADDRESS
1795  005030  012567  000012            MOV     (5)+,ACNVB      ;GET ASCII ADDRESS
1796  005034  004767  000052            JSR     %7,ACNV         ;CONVERT TO ASCII
1797  005040  004567  000122            JSR     %5,BMOVE        ;MOVE 6 CHARS TO ASCII ADDRESS
1798  005044  005102                    A1ST
1799  005046  000000                    ACNVB:  OPEN
1800  005050  000006                    6
1801  005052  000205                    RTS     %5              ;EXIT
1802  005054  012500                    ACNV4:  MOV     (5)+,%0         ;CONVERT TO 4 ASCII, GET OCTAL ADDRESS
1803  005056  012567  000012            MOV     (5)+,ACNVC      ;GET ASCII ADDRESS
1804  005062  004767  000024            JSR     %7,ACNV         ;CONVERT TO ASCII
1805  005066  004567  000074            JSR     %5,BMOVE        ;MOVE 4 CHARS TO ASCII ADDRESS.
1806  005072  005104                    A1ST+2
1807  005074  000000                    ACNVC:  OPEN
1808  005076  000004                    4
1809  005100  000205                    RTS     %5              ;EXIT
1810  005102  000000                    A1ST:   OPEN
1811  005104  000000                    OPEN
1812  005106  000000                    OPEN
1813  005110  000000                    ACNVX:  OPEN
1814  005112  012701  005110            ACNV:   MOV     #A1ST+6,%1      ;ADDR TO STORE ASCII TO R1
1815  005116  012702  000006            MOV     #6,%2           ;6 TO R2
1816  005122  011067  177762            MOV     @%0,ACNVX       ;OCTAL WORD TO ACNVX
1817  005126  016703  177756            ACNVM:  MOV     ACNVX,%3
1818  005132  042703  177770            BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT OCTAL #
1819  005136  062703  000060            ADD     #60,%3          ;ADD 60 TO CONVERT TO ASCII
1820  005142  110341                    MOVB    %3,-(1)         ;STORE ASCII BYTE
1821  005144  006067  177740            ROR     ACNVX           ;MOVE NEXT OCTAL DIGIT TO LEAST
1822  005150  006067  177734            ROR     ACNVX           ;SIGNIFICANT POSITION
1823  005154  006067  177730            ROR     ACNVX
1824  005160  005302                    DEC     %2              ;DONE 6 TIMES?
1825  005162  001361                    BNE     ACNVM           ;NO. REPEAT.
1826  005164  000207                    RTS     %7              ;YES. EXIT.
1827                                    ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1828  005166  012501                    BMOVE:  MOV     (5)+,%1         ;GET"FROM"ADDRESS
```

```
 1029  005170  012502                        MOV     (5)+,%2         ;GET"TO"ADDRESS
 1030  005172  012503                        MOV     (5)+,%3         ;GET COUNT
 1031  005174  112122                BMOVA:  MOVB    (1)+,(2)+       ;MOVE BYTE
 1032  005176  005303                        DEC     %3              ;DECREMENT COUNT
 1033  005200  001375                        BNE     BMOVA           ;BRANCH IF NOT DONE.
 1034  005202  000205                        RTS     %5              ;DONE EXIT
 1035                                ;SUBROUTINE TO CHECK FOR PUNCH READY.
 1036  005204  005777  174004        CPRDY:  TST     @PPS            ;TEST FOR ERROR BIT.
 1037  005210  100404                        BMI     CPRDYA          ;BRANCH IF ERROR BIT SET.
 1038  005212  105777  173776                TSTB    @PPS            ;TEST FOR READY BIT.
 1039  005216  100001                        BPL     CPRDYA          ;BRANCH IF READY NOT SET.
 1040  005220  000207                        RTS     %7              ;OK, EXIT.
 1041  005222  104004                CPRDYA: TYPES                   ;TYPE NOT READY MESSAGE.
 1042  005224  017456                        $M3
 1043  005226  016620                        IM16
 1044  005230  177777                        -1
 1045  005232  104010                        CHALT
 1046  005234  000763                        BR      CPRDY
 1047                                ;SUBROUTINE TO PUNCH ON H. S. PUNCH CHARACTER IN REG 0.
 1048  005236  004767  177742        HSPCH:  JSR     %7,CPRDY        ;GO CHECK FOR PUNCH READY.
 1049  005242  010077  173750                MOV     %0,@PPB         ;LOAD PUNCH BUFFER.
 1050  005246  105777  173742                TSTB    @PPS            ;WAIT FOR DONE.
 1051  005252  100375                        BPL     .-4
 1052  005254  000207                        RTS     %7              ;DONE, EXIT.
 1053                                ;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
 1054  005256  012700  015256        BDCNV:  MOV     #DECVAL,%0      ;SET UP ADDR TO STORE DECIMAL ASCII IN R0
 1055  005262  013501                        MOV     @(5)+,%1        ;BINARY VALUE TO R1.
 1056  005264  012702  005364                MOV     #ADTENP,%2      ;ADDR OF TEN POWER STRING TO R2.
 1057  005270  012767  000005  000060        MOV     #5,CNVCTR       ;SET UP FOR 5 POWER CONVERSIONS.
 1058  005276  012267  000060        BDCNVA: MOV     (2)+,TENPWR     ;MOVE POWER OF TEN VALUE TO TENPWR.
 1059  005302  004767  000010                JSR     %7,SUBTEN       ;PERFORM CONVERSION
 1060  005306  005367  000044                DEC     CNVCTR          ;DONE 5 CONVERSIONS?
 1061  005312  001371                        BNE     BDCNVA          ;BRANCH IF NOT YET 5.
 1062  005314  000205                        RTS     %5              ;YES, EXIT.
 1063  005316  005067  000036        SUBTEN: CLR     DIGIT           ;CLEAR DIGIT
 1064  005322  166701  000034        SUBTNA: SUB     TENPWR,%1       ;SUBTRACT TEN POWER FROM BINARY VALUE.
 1065  005326  103403                        BCS     SUBTNB          ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
 1066  005330  005267  000024                INC     DIGIT
 1067  005334  000772                        BR      SUBTNA
 1068  005336  066701  000020        SUBTNB: ADD     TENPWR,%1       ;RESTORE SUBTRACTED VALUE.
 1069  005342  062767  000060  000010        ADD     #60,DIGIT       ;CONVERT (DIGIT) TO ASCII
 1070  005350  116720  000004                MOVB    DIGIT,(0)+      ;MOVE ASCII CHAR TO DECVAL FIELD.
 1071  005354  000207                        RTS     %7              ;EXIT.
 1072  005356  000000        CNVCTR: OPEN
 1073  005360  000000        DIGIT:  OPEN
 1074  005362  000000        TENPWR: OPEN
 1075  005364  023420        ADTENP: 10000.
 1076  005366  001750                10000.
 1077  005370  000144                100.
 1078  005372  000012                10.
 1079  005374  000001                1
 1080
```

```
1881
1882                                        .SBTTL  PRG0 - READER LOGIC TESTS
1883                                ;PRG0 - READER LOGIC TESTS
1884                                ;
1885  005376  012767  005432  173646  PRG0:   MOV     #AT0,KSTART     ;ADDR OF 1ST ROUTINE TO KSTART.
1886  005404  005767  174430          TST     XORFLG
1887  005410  001402                  BEQ     10
1888  005412  000167  174272          JMP     GETRDY
1889  005416  104003                  10:     TYPE                    ;TYPE TITLE
1890  005420  015540                          IM0
1891  005422  004767  007614                  JSR     %7,SWTL
1892  005426  000167  174256                  JMP     GETRDY          ;GO GET STARTED.
1893                                ;*****************************************************************
1894  005432  000000                  AT0:    0                       ;TEST #
1895  005434  005462                          AT1                     ;NEXT TEST ADDR
1896  005436  001750                          1000.                   ;I COUNT
1897  005440  005450                          AT0A                    ;SCOPE ENTRY
1898                                ;*****************************************************************
1899                                ;TEST ABILITY TO REFERENCE THE READER STATUS WORD
1900  005442  012767  005456  172334          MOV     #AT0E,MACHER    ;SET UP MACHINE ERROR TRAP.
1901  005450  005777  173534          AT0A:   TST     @PRS            ;REFERENCE READER STATUS WORD.
1902  005454  104013                          SCOPE
1903  005456  104006                  AT0E:   ERROR                   ;ERROR, TRAPPED WHEN REFERENCING READER
1904  005460  104013                          SCOPE                   ;STATUS WORD (PRS).
1905                                ;*****************************************************************
1906  005462  000001                  AT1:    1                       ;TEST #
1907  005464  005512                          AT2                     ;NEXT TEST
1908  005466  001750                          1000.                   ;I COUNT
1909  005470  005500                          AT1A                    ;SCOPE ENTRY
1910                                ;*****************************************************************
1911                                ;TEST ABILITY TO REFERENCE THE READER BUFFER.
1912  005472  012767  005506  172304          MOV     #AT1E,MACHER    ;SET UP MACHINE ERROR TRAP.
1913  005500  005777  173506          AT1A:   TST     @PRB            ;REFERENCE READER BUFFER
1914  005504  104013                          SCOPE
1915  005506  104006                  AT1E:   ERROR                   ;ERROR, TRAPPED WHEN REFERENCING
1916  005510  104013                          SCOPE                   ;READER BUFFER, (PRB)
1917                                ;*****************************************************************
1918  005512  100002                  AT2:    2+MANUAL                ;TEST #
1919  005514  005562                          AT3                     ;NEXT TEST
1920  005516  001750                          1000.                   ;I COUNT
1921  005520  005546                          AT2A                    ;SCOPE ENTRY.
1922                                ;*****************************************************************
1923                                ;TEST THAT READER POWER OFF SETS ERROR BIT  (BIT 15) IN READER STATUS WORD.
1924  005522  004567  175026                  JSR     %5,PCSIM        ;PC11 SIMULATOR FOR XOR TESTER
1925  005526  000033                          33
1926  005530  005546                          AT2A                    ;ENTER IF XOR TESTER
1927                                                                  ;GO TO TYPE8 IF NOT TESTER
1928  005532  104004                          TYPE8                   ;TYPE TURN READER POWER OFF.
1929  005534  015722                          IM1
1930  005536  015756                          IM2
1931  005540  017025                          IM23
1932  005542  177777                          -1
1933  005544  000000                          HALT                    ;WAIT FOR USER
1934  005546  022777  100000  173434  AT2A:   CMP     #BIT15,@PRS     ;TEST FOR ERROR BIT ONLY.
1935  005554  001401                          BEQ     .+4             ;BRANCH IF ERROR BIT ONLY SET.
1936  005556  104006                          ERROR                   ;ERROR,WITH READER POWER OFF ONLY THE ERROR
```

```
1937  005560  104013                        $SCOPE                  ;BIT SHOULD HAVE BEEN SET,
1938                                                                ;EXAMINE READER STATUS WORD MANUALLY,
1939                                ;*****************************************************************
1940  005562  100003                AT3:    3$MANUAL                ;TEST 3
1941  005564  005630                        AT4                     ;NEXT TEST
1942  005566  001750                        1000.                   ;I COUNT
1943  005570  005616                        AT3A                    ;SCOPE ENTRY
1944                                ;*****************************************************************
1945                                ;TEST THAT READER OFF-LINE SETS ERROR BIT (BIT 15) IN READER STATUS WORD,
1946  005572  004567  174756                JSR     %5,PCSIM
1947  005576  000033                        33
1948  005600  005615                        AT3A
1949  005602  104004                        TYPE$                   ;TYPE: *TURN READER POWER ON,
1950  005604  015722                        IM1                     ;OFF-LINE, NO TAPE
1951  005606  016014                        IM3
1952  005610  017025                        IM23
1953  005612  177777                        -1
1954  005614  000000                        HALT                    ;WAIT FOR USER,
1955  005616  005777  173366        AT3A:   TST     @PRS            ;CHECK BIT 15 OF PRS
1956  005622  100401                        BMI     .+4             ;BRANCH IF BIT 15 SET,
1957  005624  104006                        ERROR                   ;ERROR, ERROR BIT(BIT15) NOT SET BY
1958  005626  104013                        $SCOPE                  ;READER BEING OFF-LINE,
1959                                ;*****************************************************************
1960  005630  100004                AT4:    4$MANUAL                ;TEST 4
1961  005632  005704                        AT5                     ;NEXT WORD
1962  005634  001750                        1000.                   ;I COUNT
1963  005636  005672                        AT4A                    ;SCOPE ENTRY
1964                                ;*****************************************************************
1965                                ;TEST THAT READER OUT OF TAPE SETS ERROR BIT(BIT 15) IN READER STATUS WORD,
1966  005640  004567  174710                JSR     %5,PCSIM
1967  005644  000033                        33
1968  005646  005664                        18
1969  005650  104004                        TYPE$                   ;TYPE: SET READER AS FOLLOWS: POWER ON ON-LINE,
1970  005652  015722                        IM1                     ;NO TAPE,
1971  005654  016051                        IM4
1972  005656  017025                        IM23
1973  005660  177777                        -1
1974  005662  000000                        HALT                    ;WAIT FOR USER,
1975  005664  005277  173320        18:     INC     @PRS            ;ENABLE READER
1976  005670  104400                        DELAYX                  ;WAIT A WHILE,
1977  005672  005777  173312        AT4A:   TST     @PRS            ;CHECK BIT 15 OF PRS
1978  005676  100401                        BMI     .+4             ;BRANCH IF BIT 15 SET,
1979  005700  104006                        ERROR                   ;ERROR, ERROR BIT (BIT 15) NOT SET BY
1980  005702  104013                        $SCOPE                  ;READER OUT OF TAPE,
1981                                ;*****************************************************************
1982  005704  100005                AT5:    5$MANUAL                ;TEST 5
1983  005706  005760                        AT6                     ;NEXT TEST
1984  005710  001750                        1000.                   ;I COUNT
1985  005712  005746                        AT5A                    ;SCOPE ENTRY
1986                                ;*****************************************************************
1987                                ;TEST THAT ERROR BIT (BIT 15) OF READER STATUS WORD (PRS) IS NOT SET
1988                                ;WITH READER POWER ON, READER ON-LINE AND WITH TAPE LOADED IN READER
1989  005714  004567  174634                JSR     %5,PCSIM
1990  005720  000433                        433                     ;TURN OFF RDR ERROR ON XOR TESTER
1991  005722  005740                        18
1992  005724  104004                        TYPE$                   ;TYPE, SET READER AS FOLLOWS: POWER ON, ON-LINE,
```

```
1993  005726  015722                        IM1                     ;TAPE IN READER.
1994  005730  016271                        IM5
1995  005732  017025                        IM23
1996  005734  177777                        -1
1997  005736  000000                        HALT                    ;WAIT FOR USER
1998  005740  005277  173244          18:   INC     @PR8              ;ENABLE READER.
1999  005744  104400                        DELAYX                  ;WAIT A WHILE.
2000  005746  005777  173236          AT5A: TST     @PR8              ;CHECK BIT 15 OF PR8
2001  005752  100001                        BPL     .+4               ;BR IF BIT 15 NOT SET.
2002  005754  104006                        ERROR                   ;ERROR, ERROR BIT (BIT 15) SET WITH NO
2003  005756  104013                        SCOPE                   ;ERROR CONDITION PRESENT.
2004                                  ;*************************************************************************
2005  005760  000006                  AT6:  6                       ;TEST 6
2006  005762  006042                        AT7                     ;NEXT TEST
2007  005764  001750                        1000.                   ;I COUNT
2008  005766  005776                        AT6A                    ;SCOPE ENTRY
2009                                  ;*************************************************************************
2010                                  ;TEST ABILITY TO SET AND CLEAR THE ID BIT (INTERRUPT ENABLE (BIT 6))
2011                                  ;IN READER STATUS WORD
2012  005770  012767  000340  172000        MOV     #PRTY7,PSW      ;SET PRIORITY 7.
2013  005776  052777  000100  173204  AT6A: BIS     #BIT6,@PR8      ;SET ID BIT (BIT 6) IN READER PR8
2014  006004  032777  000100  173176        BIT     #BIT6,@PR8      ;CHECK ID BIT IN PR8
2015  006012  001002                        BNE     AT6B            ;ID BIT SET?
2016  006014  104006                  AT6E1: ERROR                   ;NO, ERROR, FAILED TO SET ID BIT (BIT 6)
2017  006016  104013                        SCOPE                   ;IN PR8.
2018  006020  042777  000100  173162  AT6B: BIC     #BIT6,@PR8      ;CLEAR ID BIT IN PR8.
2019  006026  032777  000100  173154        BIT     #BIT6,@PR8      ;CHECK ID BIT IN PR8
2020  006034  001401                        BEQ     .+4             ;BR IF BIT NOT SET.
2021  006036  104006                        ERROR                   ;ERROR, ID BIT IN PR8 FAILED TO CLEAR.
2022  006040  104013                        SCOPE
2023                                  ;*************************************************************************
2024  006042  000007                  AT7:  7                       ;TEST 8
2025  006044  006104                        AT10                    ;NEXT TEST
2026  006046  000144                        100.                    ;I COUNT
2027  006050  006060                        AT7A                    ;SCOPE ENTRY
2028                                  ;*************************************************************************
2029                                  ;TEST ABILITY TO CLEAR ID BIT (BIT 6) WITH RESET INSTRUCTION
2030  006052  012767  000340  171716        MOV     #PRTY7,PSW      ;SET PRIORITY 7
2031  006060  052777  000100  173122  AT7A: BIS     #BIT6,@PR8      ;SET ID BIT IN PR8
2032  006066  104002                        $RESET                  ;RESET
2033  006070  032777  000100  173112        BIT     #BIT6,@PR8      ;TEST ID BIT
2034  006076  001401                        BEQ     .+4             ;BR IF IE BIT IS NOT SET.
2035  006100  104006                        ERROR                   ;ERROR, RESET INSTRUCTION FAILED TO
2036  006102  104013                        SCOPE                   ;CLEAR ID BIT IN READER PR8.
2037                                  ;*************************************************************************
2038  006104  000010                  AT10: 10                      ;TEST 9
2039  006106  006140                        AT11                    ;NEXT TEST
2040  006110  000144                        100.                    ;I COUNT
2041  006112  006114                        AT10A                   ;SCOPE ENTRY
2042                                  ;*************************************************************************
2043                                  ;TEST THAT DONE BIT SETS SOMETIME AFTER READER ENABLE.
2044  006114  004767  174522          AT10A: JSR     %7,ARRDY        ;CHECK FOR READER READY
2045  006120  005277  173064                 INC     @PR8            ;ENABLE READER
2046  006124  104400                        DELAYX                  ;WAIT.
```

```
 2047  006126  105777  173056                 TSTB    @PRS            ;TEST FOR DONE (BIT 7)
 2048  006132  100401                         BMI     .+4             ;BRANCH IF DONE BIT WAS SET,,
 2049  006134  104006                 AT10E:  ERROR                   ;ERROR, SOMETIME AFTER READER
 2050  006136  104013                         SCOPE                   ;ENABLE, DONE BIT WAS NOT SET,
 2051                                 ;*************************************************************
 2052  006140  000011                 AT11:   11                      ;TEST #
 2053  006142  006174                         AT12                    ;NEXT TEST
 2054  006144  001750                         1000.                   ;I COUNT
 2055  006146  006162                         AT11A                   ;SCOPE ENTRY
 2056                                 ;*************************************************************
 2057                                 ;TEST ABILITY TO READ DONE BIT (BIT 7 OF PRS) RELIABLY
 2058  006150  004767  174466                 JSR     %7,ARRDY        ;CHECK FOR READER READY,
 2059  006154  005277  173030                 INC     @PRS            ;ENABLE READER
 2060  006160  104400                         DELAYX                  ;WAIT,
 2061  006162  105777  173022         AT11A:  TSTB    @PRS            ;TEST DONE BIT (BIT 7 OF PRS)
 2062  006166  100401                         BMI     .+4             ;BR IF DONE BIT SET,
 2063  006170  104006                         ERROR                   ;ERROR, DONE BIT NOT SET, OR FAILED
 2064  006172  104013                         SCOPE                   ;TO READ IT,
 2065                                 ;*************************************************************
 2066  006174  000012                 AT12:   12                      ;TEST #
 2067  006176  006246                         AT13                    ;NEXT TEST
 2068  006200  000144                         100.                    ;I COUNT
 2069  006202  006204                         AT12A                   ;SCOPE ENTRY,
 2070                                 ;*************************************************************
 2071                                 ;TEST THAT RESET COMMAND CLEARS DONE BIT (BIT 7 OF PRS)
 2072  006204  004767  174432         AT12A:  JSR     %7,ARRDY        ;CHECK FOR READER READY
 2073  006210  005277  172774                 INC     @PRS            ;ENABLE READER
 2074  006214  104400                         DELAYX                  ;WAIT,
 2075  006216  105777  172766                 TSTB    @PRS            ;TEST FOR DONE BIT
 2076  006222  100005                         BPL     AT12E1          ;BRANCH IF DONE BIT NOT SET
 2077  006224  000005                         RESET                   ;RESET
 2078  006226  105777  172756                 TSTB    @PRS            ;TEST DONE BIT
 2079  006232  100403                         BMI     AT12E2          ;BRANCH IF DONE BIT STILL SET,
 2080  006234  104013                         SCOPE
 2081  006236  104006                 AT12E1: ERROR                   ;ERROR 1, DONE BIT NOT SET,
 2082  006240  104013                         SCOPE
 2083  006242  104006                 AT12E2: ERROR                   ;ERROR 2, DONE BIT NOT RESET BY
 2084  006244  104013                         SCOPE                   ;RESET INSTRUCTION,
 2085                                 ;*************************************************************
 2086  006246  000013                 AT13:   13                      ;TEST#
 2087  006250  006314                         AT14                    ;NEXT TEST
 2088  006252  000144                         100.                    ;I COUNT
 2089  006254  006256                         AT13A                   ;SCOPE ENTRY
 2090                                 ;*************************************************************
 2091                                 ;TEST THAT DONE BIT (BIT7 OF PRS) IS CLEARED WHEN ENABLING THE READER,
 2092  006256  104002                 AT13A:  $RESET                  ;RESET
 2093  006260  004767  174356                 JSR     %7,ARRDY        ;CHECK FOR READER READY
```

```
2094  006264  005277  172720          INC     @PRS            ;ENABLE READER
2095  006270  105777  172714          TSTB    @PRS            ;TEST FOR DONE BIT
2096  006274  100375                  BPL     .-4             ;BRANCH IF DONE BIT NOT SET
2097  006276  005277  172706          INC     @PRS            ;ENABLE READER AGAIN
2098  006302  105777  172702          TSTB    @PRS            ;TEST DONE BIT AGAIN
2099  006306  100001                  BPL     .+4             ;BRANCH IF DONE BIT IS RESET
2100  006310  104006                  ERROR                   ;READER ENABLE DID NOT CLEAR DONE BIT
2101  006312  104013                  SCOPE
2102                          ;*****************************************************************
2103  006314  000014          AT14:   14                      ;TEST #
2104  006316  006376                  AT15                    ;NEXT TEST
2105  006320  000144                  100.                    ;ICOUNT
2106  006322  006324                  AT14A                   ;SCOPE ENTRY
2107                          ;*****************************************************************
2108                          ;TEST THAT DONE BIT IS CLEARED BY REFERENCING READER BUFFER (PRB)
2109  006324  004767  174312  AT14A: JSR     %7,ARRDY        ;CHECK FOR READER READY.
2110  006330  005277  172654          INC     @PRS            ;ENABLE READER
2111  006334  105777  172650          TSTB    @PRS            ;TEST FOR DONE BIT
2112  006340  100375                  BPL     .-4             ;BRANCH IF DONE BIT NOT SET.
2113  006342  005777  172644          TST     @PRB            ;REFERENCE READER BUFFER (PRB)
2114  006346  105777  172636          TSTB    @PRS            ;TEST FOR DONE BIT
2115  006352  100001                  BPL     .+4             ;BR IF DONE BIT IS NOT SET.
2116  006354  104006                  ERROR                   ;ERROR 1. DONE BIT WAS NOT CLEARED
2117  006356  004567  174172          JSR     %5,PCSIM        ;GO TO PC11 XOR SIMULATOR
2118  006362  000433          XCT:    433                     ;XOR CONN.
2119  006364  006366                  AT14C                   ; RETURN ARGUMENT
2120  006366  062767  001000  177766 AT14C: ADD     #1000,XCT       ;
2121  006374  104013                  SCOPE                   ;BY REFERENCING READER BUFFER.
2122                          ;*****************************************************************
2123  006376  000015          AT15:   15                      ;TEST #
2124  006400  006446                  AT16                    ;NEXT TEST
2125  006402  000144                  100.                    ;I COUNT
2126  006404  006406                  AT15A                   ;SCOPE ENTRY
2127                          ;*****************************************************************
2128                          ;TEST THAT ENABLING READER (BIT 0 OF PRS) SETS THE BUSY BIT (BIT 11 OF PRS)
2129  006406  104002          AT15A:  SRESET
2130  006410  004767  174114          JSR     %7,ARDER        ;CHECK THAT NO READER ERROR EXISTS.
2131  006414  005277  172570          INC     @PRS
2132  006420  105777  172564          TSTB    @PRS
2133  006424  100375                  BPL     .-4
2134  006426  005277  172556          INC     @PRS            ;ENABLE READER
2135  006432  032777  004000  172550  BIT     #BIT11,@PRS     ;TEST FOR BUSY BIT
2136  006440  001001                  BNE     .+4             ;BRANCH IF BUSY BIT  SET
2137  006442  104006          AT15E:  ERROR                   ;ERROR, READER ENABLE FAILED TO SET
2138                                                          ;BUSY BIT, OR UNABLE TO READ BUSY BIT
2139  006444  104013                  SCOPE
2140                          ;*****************************************************************
2141  006446  000016          AT16:   16                      ;TEST #
2142  006450  006530                  AT17                    ;NEXT TEST
2143  006452  000144                  100.                    ;I COUNT
2144  006454  006456                  AT16A                   ;SCOPE ENTRY
2145                          ;*****************************************************************
2146                          ;TEST ABILITY TO READ BUSY BIT (BIT 11 OF PRS) RELIABLY
2147  006456  104002          AT16A:  SRESET
2148  006460  004767  174044          JSR     %7,ARDER        ;CHECK THAT NO READER ERROR EXISTS.
2149  006464  012700  000012          MOV     #10.,%0         ;SET UP COUNTER TO 10.
```

```
2159  006470  005277  172514              INC     @PRS            ;ENABLE READER
2151  006474  105777  172510              TSTB    @PRS            ;WAIT FOR DONE BIT
2152  006500  100375                      BPL     .-4
2153  006502  005277  172502              INC     @PRS            ;ENABLE READER
2154  006506  032777  004000  172474 AT16B: BIT    #BIT11,@PRS     ;TEST BUSY BIT
2155  006514  001403                      BEQ     AT16E           ;BRANCH IF BIT NOT SET
2156  006516  005300                      DEC     R0              ;DECREMENT COUNTER
2157  006520  001372                      BNE     AT16B           ;REPEAT CHECK OF BUSY BIT IF NOT 0
2158  006522  104013                      SCOPE
2159  006524  104006              AT16E:  ERROR                   ;ERROR, BUSY BIT NOT SET, OR FAILED
2160  006526  104013                      SCOPE                   ;TO READ BUSY BIT
2161                                ;*****************************************************************************
2162  006530  000017              AT17:   17                      ;TEST 0
2163  006532  006630                      AT20                    ;NEXT TEST
2164  006534  000144                      100.                    ;I COUNT
2165  006536  006540                      AT17A                   ;SCOPE ENTRY
2166                                ;*****************************************************************************
2167                                ;TEST ABILITY TO READ READER BUFFER RELIABLY.
2168  006540  012700  000144      AT17A:  MOV     #100.,R0        ;SET COUNT TO 100 IN R0
2169  006544  004767  174112              JSR     R7,AREAD        ;GET CHARACTER
2170  006550  017767  172436  172624      MOV     @PRB,CHR1       ;C(PRB) TO CHR1
2171  006556  017767  172430  172620 AT17B: MOV     @PRB,CHR2       ;C(PRB) TO CHR2
2172  006564  026767  172612  172612      CMP     CHR1,CHR2       ;COMPARE CHR1 AND CHR2,
2173  006572  001003                      BNE     AT17E           ;BRANCH IF R1 AND R2 DON'T MATCH
2174  006574  005300                      DEC     R0
2175  006576  001367                      BNE     AT17B
2176  006600  104013                      SCOPE
2177  006602  004567  176246      AT17E:  JSR     R5,ACNV4        ;CORRECT 1ST READ DATA TO ASCII
2178  006606  001402                      CHR1
2179  006610  017727                      ORGRD
2180  006612  004567  176236              JSR     R5,ACNV4
2181  006616  001404                      CHR2
2182  006620  017742                      SUBRD
2183  006622  104007                      ERROR1                  ;ERROR, REREAD OF PRB DID NOT MATCH
2184  006624  017675                      EM2                     ;INITIAL DATA READ FROM PRB,
2185  006626  104013                      SCOPE
2186                                ;*****************************************************************************
2187  006630  000020              AT20:   20                      ;TEST 8
2188  006632  006744                      AT21                    ;NEXT TEST
2189  006634  000020                      20                      ;I COUNT
2190  006636  006650                      AT20A                   ;SCOPE ENTRY
2191                                ;*****************************************************************************
2192  006640  105277  172344              INCB    @PRS
2193  006644  104000                      DELAY
2194  006646  000001                      1
2195  006650  005767  173164      AT20A:  TST     XORFLG          ;THE INSTRUCTIONS WITHIN THIS TEST
2196  006654  100031                      BPL     AT20X           ;ARE USED WITH XOR TESTER ONLY
2197  006656  013746  000004              MOV     @#4,-(SP)       ;ERRORS WILL BE INDICATED ON XOR TESTER ONLY
2198  006662  012737  006742  000004      MOV     #XTP,@#4
2199  006670  012737  000033  177060 AT20B: MOV     #33,@#177060
2200  006676  005777  172306              TST     @PRS
2201  006702  104000                      DELAY
2202  006704  000010                      10
2203
2204
2205  006706  005777  172276              TST     @PRS
```

```
2206  006712  012737  000433  177060          MOV     #433,@#177060
2207  006720  005777  172264                  TST     @PRS
2208  006724  104000                          DELAY
2209  006726  000010                          10
2210  006730  005777  172254                  TST     @PRS
2211  006734  012637  000004                  MOV     (SP)+,@#4
2212  006740  104013                  AT20X:  SCOPE
2213  006742  000002                  XTP:    RTI
2214                                  ;*****************************************************************************
2215  006744  000021                  AT21:   21                      ;TEST #
2216  006746  007014                          AT22                    ;NEXT TEST
2217  006750  000144                          100.                    ;I COUNT
2218  006752  006760                          AT21A                   ;SCOPE ENTRY
2219                                  ;*****************************************************************************
2220                                  ;TEST THAT READER IS ABLE TO INTERRUPT, IF INTERRUPT IS SERVICED, IT WILL
2221                                  ;HAVE OCCURRED AT CORRECT VECTOR.
2222  006754  104011                          STRDRV                  ;SET UP READER INTERRUPT VECTOR
2223  006756  007012                          AT21B
2224  006760  012767  000000  171010  AT21A:  MOV     #PRTY0,PSW      ;SET PROCESSOR PRIORITY TO 0
2225  006766  042777  000100  172214          BIC     #BIT6,@PRS      ;DISABLE READER INTERRUPT.
2226  006774  004767  173662                  JSR     %7,AREAD        ;GO READ CHARACTER.
2227  007000  052777  000100  172202          BIS     #BIT6,@PRS      ;ENABLE READER INTERRUPT.
2228  007006  000240                          NOP                     ;NO OP
2229  007010  104006                  AT21E:  ERROR                   ;ERROR, READER FAILED TO INTERRUPT.
2230  007012  104013                  AT21B:  SCOPE
2231                                  ;*****************************************************************************
2232  007014  000022                  AT22:   22                      ;TEST #
2233  007016  007070                          AT23                    ;NEXT TEST
2234  007020  000144                          100.                    ;I COUNT
2235  007022  007030                          AT22A                   ;SCOPE ENTRY
2236                                  ;*****************************************************************************
2237                                  ;TEST THAT READER DOES NOT INTERRUPT WITH PROCESSOR AT SAME PRIORITY
2238                                  ;LEVEL AS READER.
2239  007024  104011                          STRDRV                  ;SET UP READER INTERRUPT VECTOR
2240  007026  007064                          AT22E
2241  007030  016767  172166  170740  AT22A:  MOV     RDRLVL,PSW      ;SET PROCESSOR PRIORITY SAME AS READER PRIORITY.
2242  007036  005077  172146                  CLR     @PRS            ;DISABLE READER INTERRUPT.
2243  007042  004767  173614                  JSR     %7,AREAD        ;GO READ A CHARACTER.
2244  007046  052777  000100  172134          BIS     #BIT6,@PRS      ;ENABLE READER INTERRUPT.
2245  007054  000240                          NOP                     ;OK IF NO INTERRUPT OCCURS.
2246  007056  005077  172126                  CLR     @PRS            ;DISABLE READER INTERRUPT.
2247  007062  104013                          SCOPE
2248  007064  104006                  AT22E:  ERROR                   ;ERROR, READER ERRONEOUSLY INTERRUPTED
2249                                                                  ;WITH PROCESSOR AT SAME PRIORITY LEVEL AS
2250                                                                  ;THE READER, OR THE READER IS AT HIGHER
2251  007066  104013                          SCOPE                   ;PRIORITY LEVEL THAN SPECIFIED AT RDRLVL.
2252                                  ;*****************************************************************************
2253  007070  000023                  AT23:   23                      ;TEST #
2254  007072  007146                          AT24                    ;NEXT TEST
2255  007074  000144                          100.                    ;I COUNT
2256  007076  007104                          AT23A                   ;SCOPE ENTRY
2257                                  ;*****************************************************************************
2258                                  ;TEST THAT READER INTERRUPTS WITH PROCESSOR AT PRIORITY 1 LEVEL LOWER
2259                                  ;THAN READER'S
2260  007100  104011                          STRDRV                  ;SET UP READER INTERRUPT VECTOR
2261  007102  007144                          AT23B
```

```
2262  007104  016767  172112  170664  AT23A:  MOV     RDRLVL,PSW      ;SET PROCESSOR PRIORITY ONE LEVEL LOWER
2263  007112  162767  000040  170656          SUB     #40,PSW         ;THAN READER PRIORITY
2264  007120  042777  000100  172062          BIC     #BIT6,@PRS      ;DISABLE READER INTERRUPT
2265  007126  004767  173530                  JSR     %7,AREAD        ;GO READ A CHARACTER.
2266  007132  052777  000100  172050          BIS     #BIT6,@PRS      ;ENABLE READER INTERRUPT
2267  007140  000240                          NOP                     ;NOP
2268  007142  104006                  AT23E:  ERROR                   ;READER FAILED TO INTERRUPT WITH
2269                                                                  ;PROCESSOR PRIORITY ONE LEVEL LOWER THAN
2270                                                                  ;READER.  THEREFORE, READER PRIORITY MUST BE
2271  007144  104013                  AT23B:  SCOPE                   ;LOWER THAN SPECIFIED AT RDRLVL
2272                                  ;*****************************************************************************
2273  007146  000024                  AT24:   24                      ;TEST #
2274  007150  007240                          AT25                    ;NEXT TEST
2275  007152  000144                          100.                    ;I COUNT
2276  007154  007156                          AT24A                   ;SCOPE ENTRY
2277                                  ;*****************************************************************************
2278                                  ;TEST THAT READER DOES NOT REINTERRUPT AFTER RTI WHEN DONE BIT IS NOT CLEARED
2279  007156  104011                  AT24A:  STRDRV                  ;SET READER INTERRUPT VECTOR
2280  007160  007214                          AT24C
2281  007162  012767  000000  170606          MOV     #PRTY0,PSW      ;SET PROCESSOR TO PRIORITY 0
2282  007170  005077  172014                  CLR     @PRS            ;DISABLE READER INTERRUPT.
2283  007174  004767  173462                  JSR     %7,AREAD        ;GO READ A CHARACTER.
2284  007200  052777  000100  172002          BIS     #BIT6,@PRS      ;ENABLE READER INTERRUPT
2285  007206  000240                          NOP
2286  007210  104006                  AT24E1: ERROR                   ;ERROR 1,  READER FAILED TO INTERRUPT
2287  007212  104013                          SCOPE
2288  007214  012777  007234  171776  AT24C:  MOV     #AT24E2,@RDRVTR ;CHANGE INTERRUPT VECTOR TO AT24E2
2289  007222  012716  007230                  MOV     #AT24D,(SP)
2290  007226  000002                          RTI                     ;RETURN FROM INTERRUPT
2291  007230  000240                  AT24D:  NOP
2292  007232  104013                          SCOPE
2293  007234  104006                  AT24E2: ERROR                   ;ERROR 2, READER REINTERRUPTED AFTER
2294  007236  104013                          SCOPE                   ;RTI WITH DONE BIT LEFT ON
2295                                  ;*****************************************************************************
2296  007240  000025                  AT25:   25                      ;TEST #
2297  007242  007316                          AT26                    ;NEXT TEST
2298  007244  001750                          1000.                   ;I COUNT
2299  007246  007254                          AT25A                   ;SCOPE ENTRY.
2300                                  ;*****************************************************************************
2301                                  ;TEST THAT READER INTERRUPTS IMMEDIATELY UPON LOWERING CP PRIORITY TO 0.
2302  007250  104011                          STRDRV                  ;SET READER INTERRUPT VECTOR TO
2303  007252  007314                          AT25B                   ;AT27B.
2304  007254  012767  000340  170514  AT25A:  MOV     #PRTY7,PSW      ;SET CP PRIORITY TO 7.
2305  007262  005077  171722                  CLR     @PRS            ;DISABLE PTRI.
2306  007266  004767  173370                  JSR     %7,AREAD        ;READ A CHARACTER.
2307  007272  052777  000100  171710          BIS     #BIT6,@PRS      ;ENABLE PTRI
2308  007300  005067  170472                  CLR     PSW             ;LOWER PRIORITY TO 0.
2309  007304  012767  000340  170464          MOV     #PRTY7,PSW      ;RAISE PRIORITY BACK TO 7.
2310  007312  104006                  AT25E:  ERROR                   ;ERROR, READER FAILED TO INTERRUPT IMMEDIATELY
2311                                                                  ;AFTER LOWERING PRIORITY TO 0
2312  007314  104013                  AT25B:  SCOPE                   ;INTERRUPTS TO HERE IF SUCCESSFUL.
2313                                  ;*****************************************************************************
2314  007316  100026                  AT26:   26+MANUAL               ;TEST #
2315  007320  007412                          AT27                    ;NEXT TEST
2316  007322  000144                          100.                    ;I COUNT
2317  007324  007350                          AT26A                   ;SCOPE ENTRY
```

```
 2318                                   ;*****************************************************************************
 2319                                   ;TEST THAT READER ERROR CRIPPLES READER ENABLE
 2320  007326  004567  173222                 JSR     %5,PCSIN
 2321  007332  000033                         33
 2322  007334  007350                         AT26A                   ;SKIP THIS XOR TEST
 2323  007336  104004                         TYPES                   ;TYPE, SET READER AS FOLLOWS: POWER ON,
 2324  007340  016405                         IN10                    ;OFF-LINE, TAPE IN READER
 2325  007342  017025                         IN23
 2326  007344  177777                         -1
 2327  007346  000000                         HALT
 2328  007350  005777  171634         AT26A:  TST     @PRS            ;CHECK FOR ERROR BIT.
 2329  007354  100012                         BPL     AT26E1          ;BRANCH IF ERROR BIT NOT SET.
 2330  007356  005277  171626                 INC     @PRS            ;ATTEMPT READER ENABLE
 2331  007362  005767  172452                 TST     XORFLG
 2332  007366  001010                         BNE     AT26B
 2333  007370  032777  004000  171612         BIT     #BIT11,@PRS     ;TEST READER BUSY BIT
 2334  007376  001003                         BNE     AT26E2
 2335  007400  104013                         SCOPE
 2336  007402  104006         AT26E1: ERROR                           ;ERROR 1, ERROR BIT NOT SET, OR READER
 2337  007404  104013                         SCOPE                   ;NOT SET UP AS SPECIFIED.
 2338  007406  104006         AT26E2: ERROR                           ;READER ENABLE WITH ERROR CONDITION SET
 2339                                                                 ;BUSY BIT, ERROR CONDITION SHOULD HAVE
 2340  007410  104013         AT26B:  SCOPE                           ;DISABLED READER ENABLE.
 2341                                   ;*****************************************************************************
 2342  007412  100027         AT27:   27.+MANUAL                      ;TEST #
 2343  007414  007530                 AT30                            ;NEXT TEST.
 2344  007416  000144                 100.                            ;I COUNT
 2345  007420  007444                 AT27A                           ;SCOPE ENTRY
 2346                                   ;*****************************************************************************
 2347                                   ;TEST THAT ERROR BIT IS ABLE TO INTERRUPT, AND AFTER INTERRUPT
 2348                                   ;SERVICE IT DOES NOT REINTERRUPT AGAIN.
 2349  007422  004567  173126                 JSR     %5,PCSIN
 2350  007426  000033                         33
 2351  007430  007444                         AT27A
 2352  007432  104004                         TYPES                   ;TYPE: SET READER AS FOLLOWS: POWER OFF
 2353  007434  016405                         IN10                    ;OFFLINE; TAPE IN READER
 2354  007436  017025                         IN23
 2355  007440  177777                         -1
 2356  007442  000000                         HALT
 2357  007444  104011         AT27A:  STRDRV                          ;SET UP READER INTERRUPT VECTOR
 2358  007446  007500                 AT27C
 2359  007450  005777  171534                 TST     @PRS            ;TEST ERROR BIT.
 2360  007454  100023                         BPL     AT27E1          ;BRANCH IF ERROR BIT NOT SET (BIT 15 OF PRS).
 2361  007456  042777  000100  171524         BIC     #BIT6,@PRS      ;DISABLE READER INTERRUPT.
 2362  007464  052777  000100  171516         BIS     #BIT6,@PRS      ;ENABLE READER INTERRUPT
 2363  007472  000240                         NOP
 2364  007474  104006         AT27E2: ERROR                           ;ERROR 2, ERROR CONDITION FAILED TO CAUSE
 2365                                                                 ;READER INTERRUPT
 2366  007476  104013                         SCOPE
 2367  007500  012777  007520  171512 AT27C:  MOV     #AT27E3,@RDRVTR ;SET UP READER SERVICE TO AT27E3
 2368  007506  012716  007514                 MOV     #AT27D,@6       ;MODIFY INTERRUPT RETURN ADDRESS
 2369  007512  000002                         RTI                     ;RETURN FROM INTERRUPT
 2370  007514  000240         AT27D:  NOP                             ;OK IF NO INTERRUPT.
 2371  007516  104013                         SCOPE
 2372  007520  104006         AT27E3: ERROR                           ;ERROR 3, ERROR CONDITION RESULTED IN
 2373                                                                 ;A REINTERRUPT AFTER INITIAL INTERRUPT
```

```
2374  007522  104013                          SCOPE                   ;WAS SERVICED
2375  007524  104006                AT27E1:   ERROR                   ;ERROR 1, ERROR BIT NOT SET, OR READER
2376  007526  104013                          SCOPE                   ;NOT SET UP AS SPECIFIED
2377                                ;*************************************************************************
2378  007530  100030                AT30:     30$MANUAL               ;TEST 0
2379  007532  177777                          -1                      ;LAST TEST
2380  007534  001750                          1000.                   ;I COUNT
2381  007536  007576                          AT30A                   ;SCOPE ENTRY.
2382                                ;*************************************************************************
2383                                ;TEST THAT WITH ERROR BIT SET AND HAVING GENERATED AN INTERRUPT,
2384                                ;ISSUING A READER ENABLE CAUSES AN IMMEDIATE INTERRUPT.
2385  007540  004567  173010                    JSR     %5,PCSIM
2386  007544  000033                          33
2387  007546  007562                          18
2388  007550  104004                          TYPES
2389  007552  016405                          IM10
2390  007554  017025                          IM23
2391  007556  177777                          -1
2392  007560  000000                          HALT
2393  007562  104002                18:       $RESET
2394  007564  104011                          STRDRV                  ;SET PTR VECTOR TO AT30B.
2395  007566  007624                          AT30B
2396  007570  005277  171414                  INC     @PRS            ;ENABLE READER.
2397  007574  104400                          DELAYX                  ;WAIT A WHILE.
2398  007576  005777  171406        AT30A:    TST     @PRS            ;TEST FOR ERROR.
2399  007602  100025                          BPL     AT30E1          ;BRANCH IF ERROR NOT SET.
2400  007604  005077  171400                  CLR     @PRS            ;DISABLE PTRI
2401  007610  052777  000100  171372          BIS     #BIT6,@PRS      ;ENABLE PTRI
2402  007616  000240                          NOP
2403  007620  104006                AT30E2:   ERROR                   ;ERROR FAILED TO INTERRUPT.
2404  007622  104013                          SCOPE
2405  007624  012716  007632        AT30B:    MOV     #AT30C,@#4      ;ERROR INTERRUPTS TO HERE, SET UP INTERRUPT
2406  007630  000002                          RTI                     ;EXIT, AND EXIT.
2407  007632  104011                AT30C:    STRDRV                  ;SET PTR VECTOR TO AT30D.
2408  007634  007654                          AT30D
2409  007636  005777  171346                  TST     @PRS            ;TEST THAT ERROR BIT IS STILL ON.
2410  007642  100005                          BPL     AT30E1          ;BRANCH IF NO ERROR BIT.
2411  007644  005277  171340                  INC     @PRS            ;READER ENABLE, SHOULD CAUSE
2412  007650  000240                          NOP                     ;IMMEDIATE INTERRUPT.
2413  007652  104006                AT30E3:   ERROR                   ;ERROR, READER ENABLE WITH PREVIOUS ERROR
2414                                                                  ;INTERRUPT FAILED TO INTERRUPT.
2415  007654  104013                AT30D:    SCOPE                   ;OK, INTERRUPT OCCUREED.
2416  007656  005077  171326        AT30E1:   CLR     @PRS            ;DISABLE PTRI
2417  007662  104006                          ERROR                   ;ERROR BIT NOT SET.
2418  007664  104013                          SCOPE
```

```
                                                        .SBTTL  PRG1 - READER TEST
                                                ;PRG1: READER TEST
 2421  007666  012767  007724  171356  PRG1:   MOV     #BT0,KSTART     ;SET ADDRESS OF FIRST ROUTINE
 2422  007674  104004                          TYPE$                   ;TYPE SET UP INSTRUCTIONS
 2423  007676  016360                          IM7
 2424  007700  016334                          IM6
 2425  007702  017025                          IM23
 2426  007704  177777                          -1
 2427  007706  000000                          HALT
 2428  007710  004767  005326                  JSR     %7,SWTL
 2429  007714  004767  174000                  JSR     PC,RTMCAL       ;CALIBRATE DELAY RTN WITH READER.
 2430  007720  000167  171764                  JMP     GETRDY          ;GO GET STARTED.
 2431                                  ;*****************************************************************
 2432  007724  000000                  BT0:    0                       ;TEST #
 2433  007726  007752                          BT1                     ;NEXT TEST
 2434  007730  023420                          10000.                  ;I COUNT
 2435  007732  007740                          BT0A                    ;SCOPE ENTRY
 2436                                  ;*****************************************************************
 2437                                  ;READ AND CHECK 10000 CHARACTERS OF SPECIAL BINARY COUNT PATTERN, FULL SPEED.
 2438  007734  004767  174512                  JSR     %7,BSYNC        ;SYNC READER; SET ERROR COUNTER.
 2439  007740  004767  173136          BT0A:   JSR     %7,BREAD        ;GO READ CHARACTER
 2440  007744  004767  174422                  JSR     %7,BCHECK       ;GO CHECK CHARACTER READ.
 2441  007750  104013                          SCOPE
 2442                                  ;*****************************************************************
 2443  007752  000001                  BT1:    1                       ;TEST #
 2444  007754  010010                          BT2                     ;NEXT TEST
 2445  007756  000764                          500.                    ;I COUNT
 2446  007760  007774                          BT1A                    ;SCOPE ENTRY
 2447                                  ;*****************************************************************
 2448                                  ;READ AND CHECK 500 CHARACTERS OF SPECIAL BINARY COUNT PATTERN,
 2449                                  ;RANDOM STALL BETWEEN CHARACTERS (0 TO 7 MSECS).
 2450  007762  012767  177770  174316          MOV     #177770,STLMSK
 2451  007770  004767  174456                  JSR     %7,BSYNC        ;SYNC READER; SET ERROR COUNTER
 2452  007774  104005          BT1A:   STALL                           ;RANDOM STALL (0 TO 7 MSECS)
 2453  007776  004767  173100                  JSR     %7,BREAD        ;GO READ CHARACTER
 2454  010002  004767  174364                  JSR     %7,BCHECK       ;GO CHECK CHARACTER READ
 2455  010006  104013                          SCOPE
 2456                                  ;*****************************************************************
 2457  010010  000002                  BT2:    2                       ;TEST #
 2458  010012  010062                          BT3                     ;NEXT TEST
 2459  010014  001750                          1000.                   ;I COUNT
 2460  010016  010032                          BT2A                    ;SCOPE ENTRY
 2461                                  ;*****************************************************************
 2462                                  ;READ 1000 GROUPS OF 3 CHARACTERS EACH.  RANDOM STALL (0 TO 31 MSECS) BEFORE EACH GROUP.
 2463  010020  012767  177740  174260          MOV     #177740,STLMSK  ;LIMIT STALLS TO 31 MSECS.
 2464  010026  004767  174420                  JSR     %7,BSYNC        ;SYNC READER.  SET ERROR COUNTER
 2465  010032  012767  000003  174330  BT2A:   MOV     #3,RNCNT        ;SET CHAR COUNT TO 3.
 2466  010040  104005                          STALL                   ;RANDOM STALL (0 TO 31 MSECS).
 2467  010042  004767  173034          BT2C:   JSR     %7,BREAD        ;GO READ CHARACTER.
 2468  010046  004767  174320                  JSR     %7,BCHECK       ;GO CHECK CHARACTER READ.
 2469  010052  005367  174312                  DEC     RNCNT           ;3 CHARS READ?
 2470  010056  001371                          BNE     BT2C            ;BR IF NOT 3 CHARS YET.
 2471  010060  104013                          SCOPE
 2472                                  ;*****************************************************************
 2473  010062  000003                  BT3:    3                       ;TEST #
 2474  010064  010140                          BT4                     ;NEXT TEST
```

K4 CU

```
2475  010066  001750                        1000.                   ;I COUNT
2476  010070  010112                        BT3A                    ;SCOPE ENTRY.
2477                                  ;*****************************************************************
2478                                  ;READ AND CHECK 1000 CHARACTER GROUPS OF RANDOM LENGTH (1 TO 15),
2479                                  ;RANDOM STALL (0 TO 31 MSECS) BETWEEN GROUPS.
2480  010072  012767  177740  174206        MOV     #177740,STLMSK  ;LIMIT STALLS TO 31 MSECS.
2481  010100  012767  177760  174260        MOV     #177760,RCMSK   ;LIMIT MAX CHAR COUNT TO 15 CHARS.
2482  010106  004767  174340                JSR     %7,DSYNC        ;SYNC READER.  SET ERROR COUNTER.
2483  010112  004767  174230        BT3A:   JSR     %7,GRCNT        ;GENERATE RANDOM CHAR COUNT.
2484  010116  104005                        STALL
2485  010120  004767  172756        BT3C:   JSR     %7,DREAD        ;GO READ CHARACTER.
2486  010124  004767  174242                JSR     %7,DCHECK       ;GO CHECK CHARACTER.
2487  010130  005367  174234                DEC     RNCNT           ;ALL CHARS READ?
2488  010134  001371                        BNE     BT3C            ;BRANCH IF NOT.
2489  010136  104013                        SCOPE
2490                                  ;*****************************************************************
2491  010140  000004                BT4:    4                       ;TEST #
2492  010142  177777                        -1                      ;LAST TEST
2493  010144  001750                        1000.                   ;I COUNT
2494  010146  010170                        BT4A                    ;SCOPE ENTRY
2495                                  ;*****************************************************************
2496                                  ;READ AND CHECK 1000 CHARACTER GROUPS OF SPECIAL BINARY COUNT PATTERN,
2497                                  ;RANDOM LENGTH
2498                                  ;GROUPS (BETWEEN 1 AND 77), RANDOM STALL BETWEEN GROUPS (0 TO 31 MSECS),
2499  010150  012767  177740  174130        MOV     #177740,STLMSK
2500  010156  012767  177700  174202        MOV     #177700,RCMSK
2501  010164  004767  174262                JSR     %7,DSYNC        ;SYNC READER; SET ERROR COUNTER.
2502  010170  004767  174152        BT4A:   JSR     %7,GRCNT        ;GENERATE RANDOM CHARACTER COUNT.
2503  010174  104005                        STALL                   ;RANDOM STALL (0 TO 31MSECS)
2504  010176  004767  172700        BT4C:   JSR     %7,DREAD        ;GO READ CHARATER
2505  010202  004767  174164                JSR     %7,DCHECK       ;GO CHECK CHARACTER READ
2506  010206  005367  174156                DEC     RNCNT           ;DECREMENT RANDOM CHAR COUNT
2507  010212  001371                        BNE     BT4C            ;GO READ AGAIN IF COUNT NOT 0.
2508  010214  104013                        SCOPE
```

```
 2509                                           .SBTTL  PRG2 - PUNCH LOGIC TEST8
 2510  010216  012767  010240  171026  PRG2:   MOV     #CT0,KSTART     ;ADDR OF 1ST ROUTINE TO KSTART
 2511  010224  104003                          TYPE                    ;TYPE TITLE.
 2512  010226  015574                          IM8A
 2513  010230  004767  005006                  JSR     %7,$WTL
 2514  010234  000167  171450                  JMP     GETRDY          ;GO GET STARTED.
 2515                                  ;*****************************************************************
 2516  010240  000000                 CT0:    0                       ;TEST #
 2517  010242  010270                          CT1                     ;NEXT TEST
 2518  010244  001750                          1000.                   ;I COUNT
 2519  010246  010256                          CT0A                    ;SCOPE ENTRY
 2520                                  ;*****************************************************************
 2521                                  ;TEST ABILITY TO REFERENCE THE PUNCH STATUS WORD (PPS)
 2522  010250  012767  010264  167526          MOV     #CT0E,MACHER
 2523  010256  005777  170732          CT0A:   TST     @PPS            ;REFERENCE PUNCH STATUS WORD
 2524  010262  104013                          SCOPE
 2525  010264  104006                  CT0E:   ERROR                   ;ERROR, TRAPPED WHEN REFERENCING PUNCH
 2526  010266  104013                          SCOPE                   ;STATUS WORD (PPS).
 2527                                  ;*****************************************************************
 2528  010270  000001                 CT1:    1                       ;TEST #
 2529  010272  010320                          CT2                     ;NEXT TEST
 2530  010274  001750                          1000.                   ;I COUNT
 2531  010276  010306                          CT1A                    ;SCOPE ENTRY
 2532                                  ;*****************************************************************
 2533                                  ;TEST ABILITY TO REFERENCE THE PUNCH BUFFER (PPB)
 2534  010300  012767  010314  167476          MOV     #CT1E,MACHER    ;SET UP MACHINE ERROR TRAP.
 2535  010306  005777  170704          CT1A:   TST     @PPB            ;REFERENCE PUNCH BUFFER.
 2536  010312  104013                          SCOPE
 2537  010314  104006                  CT1E:   ERROR                   ;TRAPPED WHEN REFERENCING
 2538  010316  104013                          SCOPE                   ;PUNCH BUFFER (PPB)
 2539                                  ;*****************************************************************
 2540  010320  100002                 CT2:    2+MANUAL                ;TEST #
 2541  010322  010370                          CT3                     ;NEXT TEST
 2542  010324  001750                          1000.                   ;I COUNT
 2543  010326  010354                          CT2A                    ;SCOPE ENTRY
 2544                                  ;*****************************************************************
 2545                                  ;TEST THAT PUNCH POWER OFF  SETS ERROR AND READY BITS IN PPS
 2546  010330  004567  172220                  JSR     %5,PCSIM
 2547  010334  000433                          433
 2548  010336  010354                          CT2A
 2549  010340  104004                          TYPE8                   ;TYPE INSTRUCTIONS TO TURN POWER
 2550  010342  016434                          IM11                    ;OFF AND REMOVE TAPE FROM
 2551  010344  016467                          IM12                    ;PUNCH
 2552  010346  017025                          IM23
 2553  010350  177777                          -1
 2554  010352  000000                          HALT                    ;WAIT FOR USER
 2555  010354  022777  100200  170632  CT2A:   CMP     #100200,@PPS    ;TEST PPS.
 2556  010362  001401                          BEQ     .+4             ;BRANCH IF ERROR AND READY SET.
 2557  010364  104006                          ERROR                   ;ERROR, PUNCH ERROR BIT (BIT 15) NOT SET BY
 2558  010366  104013                          SCOPE                   ;PUNCH POWER OFF, OR READY BIT NOT SET, OR
 2559                                                                  ;SOME OTHER BIT IS SET, EXAMINE PUNCH
 2560                                                                  ;STATUS WORD MANUALLY.
 2561                                  ;*****************************************************************
 2562  010370  100003                 CT3:    3+MANUAL                ;TEST #
 2563  010372  010436                          CT4                     ;NEXT TEST
 2564  010374  001750                          1000.                   ;I COUNT
```

```
2565  010376  010424                        CT3A                    ;SCOPE ENTRY
2566                                ;*****************************************************************
2567                                ;TEST THAT PUNCH OUT OF TAPE SETS ERROR BIT IN PPS
2568  010400  004567  172150                JSR     %5,PCSIM
2569  010404  000433                        433
2570  010406  010424                        CT3A
2571  010410  104004                        TYPES                   ;TYPE INSTRUCTIONS TO TURN PUNCH
2572  010412  016434                        IM11                    ;POWER ON, AND REMOVE TAPE FROM PUNCH.
2573  010414  016513                        IM13
2574  010416  017025                        IM23
2575  010420  177777                        -1
2576  010422  000000                        HALT                    ;WAIT FOR USER.
2577  010424  005777  170564        CT3A:   TST     @PPS            ;TEST PPS
2578  010430  100401                        BMI     .+4             ;BR IF ERROR BIT SET.
2579  010432  104006                        ERROR                   ;ERROR, PUNCH OUT OF TAPE FAILED TO SET
2580  010434  104013                        SCOPE                   ;THE ERROR BIT IN PPS (BIT 15).
2581                                ;*****************************************************************
2582  010436  100004                CT4:    4+MANUAL                ;TEST 4
2583  010440  010504                        CT5                     ;NEXT TEST
2584  010442  001750                        1000.                   ;I COUNT
2585  010444  010472                        CT4A                    ;SCOPE ENTRY
2586                                ;*****************************************************************
2587                                ;TEST THAT PUNCH ERROR BIT IS NOT SET WHEN PUNCH POWER IS ON AND TAPE IS IN PUNCH.
2588  010446  004567  172102                JSR     %5,PCSIM
2589  010452  000033                        33
2590  010454  010472                        CT4A
2591  010456  104004                        TYPES                   ;TYPE INSTRUCTIONS TO LOAD TAPE IN
2592  010460  016434                        IM11                    ;PUNCH AND TURN POWER ON.
2593  010462  016536                        IM14
2594  010464  017025                        IM23
2595  010466  177777                        -1
2596  010470  000000                        HALT                    ;WAIT FOR USER.
2597  010472  005777  170516        CT4A:   TST     @PPS            ;TEST PPS
2598  010476  100001                        BPL     .+4             ;BR IF ERROR BIT NOT SET.
2599  010500  104006                        ERROR                   ;ERROR, ERROR BIT SET WITH NO ERROR
2600  010502  104013                        SCOPE                   ;CONDITION PRESENT.
2601                                ;*****************************************************************
2602  010504  000005                CT5:    5                       ;TEST 5
2603  010506  010566                        CT6                     ;NEXT TEST
2604  010510  001750                        1000.                   ;I COUNT
2605  010512  010514                        CT5A                    ;SCOPE ENTRY
2606                                ;*****************************************************************
2607                                ;TEST ABILITY TO SET AND CLEAR ID BIT (BIT 6) IN PPS
2608  010514  012767  000340  167254 CT5A:   MOV     #PRTY7,PSW      ;SET PRIORITY 7
2609  010522  052777  000100  170464         BIS     #BIT6,@PPS      ;SET ID BIT IN PPS (BIT 6 )
2610  010530  032777  000100  170456         BIT     #BIT6,@PPS      ;CHECK ID BIT IN PPS
2611  010536  001002                        BNE     CT5B            ;BRANCH IF BIT SET
2612  010540  104006                        ERROR                   ;FAILED TO SET ID BIT (BIT 6) IN PPS
2613  010542  104013                        SCOPE
2614  010544  042777  000100  170442 CT5B:   BIC     #BIT6,@PPS      ;CLEAR ID BIT IN PPS
2615  010552  032777  000100  170434         BIT     #BIT6,@PPS      ;CHECK ID BIT IN PPS
2616  010560  001401                        BEQ     .+4             ;BR IF BIT IS NOT SET.
2617  010562  104006                        ERROR                   ;ERROR, ID BIT IN PPS FAILED TO CLEAR
2618  010564  104013                        SCOPE
2619                                ;*****************************************************************
2620  010566  000006                CT6:    6                       ;TEST 6
```

```
2621  010570  010630                        CT7                     ;NEXT TEST
2622  010572  000144                        100.                    ;I COUNT
2623  010574  010576                        CT6A                    ;SCOPE ENTRY
2624                                  ;*****************************************************************
2625                                  ;TEST ABILITY TO CLEAR ID BIT IN PPS (BIT6) WITH RESET INSTRUCTION
2626  010576  012767  000340  167172  CT6A:  MOV     #PRTY7,PSW      ;SET PRIORITY 7.
2627  010604  052777  000100  170402         BIS     #BIT6,@PPS      ;SET ID BIT IN PPS.
2628  010612  104002                        SRESET                  ;RESET.
2629  010614  032777  000100  170372         BIT     #BIT6,@PPS      ;TEST ID BIT IN PPS.
2630  010622  001401                        BEQ     .+4             ;BR IF IE BIT NOT SET.
2631  010624  104006                        ERROR                   ;ERROR. RESET INSTRUCTION FAILED TO
2632  010626  104013                        SCOPE                   ;CLEAR ID BIT (BIT 6) IN PPS.
2633                                  ;*****************************************************************
2634  010630  000007                  CT7:  7                       ;TEST #
2635  010632  010652                        CT10                    ;NEXT TEST
2636  010634  001750                        1000.                   ;I COUNT
2637  010636  010640                        CT7A                    ;SCOPE ENTRY
2638                                  ;*****************************************************************
2639                                  ;TEST THAT READY BIT (BIT 7) IS SET FOLLOWING A RESET INSTRUCTION, AND
2640                                  ;THAT THE READY BIT CAN BE READ RELIABLY.
2641  010640  105777  170350          CT7A:  TSTB    @PPS            ;TEST PPS
2642  010644  100401                        BMI     .+4             ;BR IF READY BIT SET.
2643  010646  104006                        ERROR                   ;ERROR. RESET FAILED TO SET READY BIT,
2644  010650  104013                        SCOPE                   ;OR FAILED TO READ READY BIT.
2645                                  ;*****************************************************************
2646  010652  000010                  CT10:  10                     ;TEST #
2647  010654  010742                        CT11                    ;NEXT TEST
2648  010656  000400                        256.                    ;I COUNT
2649  010660  010662                        CT10A                   ;SCOPE ENTRY
2650                                  ;*****************************************************************
2651                                  ;TEST THAT READY BIT (BIT 7) OF PPS IS RESET BY LOADING PUNCH BUFFER (PPB)
2652  010662  104002                  CT10A: SRESET                  ;RESET
2653  010664  004767  174314                 JSR     %7,CPRDY        ;CHECK FOR PUNCH READY
2654  010670  012777  000000  170320  CT10B: MOV     #0,@PPB         ;LOAD 0 INTO PUNCH BUFFER (PPB)
2655  010676  105777  170312                 TSTB    @PPS            ;TEST PPS
2656  010702  100001                        BPL     .+4             ;BR IF READY BIT RESET.
2657  010704  104006                        ERROR                   ;ERROR. LOADING PUNCH BUFFER (PPB)
2658  010706  013746  000004                MOV     @#4,-(%6)
2659  010712  012737  010736  000004         MOV     #XPBE,@#4
2660  010720  005737  177060                TST     @#177060
2661  010724  105237  010672                INCB    @#CT10B+2
2662  010730  012637  000004          XP:   MOV     (%6)+,@#4
2663  010734  104013                  CT10C: SCOPE                   ;FAILED TO RESET READY BIT IN PPS
2664  010736  022626                  XPBE:  CMP     (%6)+,(%6)+
2665  010740  000773                        BR      XP
2666
2667                                  ;*****************************************************************
2668  010742  000011                  CT11:  11                     ;TEST #
2669  010744  011004                        CT12                    ;NEXT TEST
2670  010746  000144                        100.                    ;I COUNT
2671  010750  010752                        CT11A                   ;SCOPE ENTRY
2672                                  ;*****************************************************************
2673                                  ;TEST THAT READY BIT (BIT 7) IS NOT RESET BY BYTE LOADING PPB+1.
2674  010752  104002                  CT11A: SRESET                  ;RESET
2675  010754  004767  174224                 JSR     %7,CPRDY        ;CHECK FOR PUNCH READY.
2676  010760  016700  170232                 MOV     PPB,%0
```

```
2677  011764  005200                        INC     %0
2678  011766  112710  000000                MOVB    #0,@%6          ;LOAD PPB+1
2679  011772  105777  170216                TSTB    @PPS            ;TEST PPS
2680  011776  100401                        BMI     .+4             ;BRANCH IF READY BIT NOT RESET.
2681  012000  104006              CT11E:    ERROR                   ;ERROR, LOADING PPB+1 CLEARED READY BIT.
2682  012002  104013                        SCOPE
2683                               ;*****************************************************************************
2684  012004  000012              CT12:     12                      ;TEST #
2685  012006  012052                        CT13                    ;NEXT TEST
2686  012010  001750                        1000.                   ;I COUNT
2687  012012  012020                        CT12A                   ;SCOPE ENTRY
2688                               ;*****************************************************************************
2689                               ;TEST THAT PUNCH (READY BIT) IS ABLE TO INTERRUPT. IF THE INTERRUPT IS
2690                               ;SERVICED, IT WILL HAVE OCCURRED AT CORRECT VECTOR.
2691  012014  104012                        STPCHV                  ;SET UP PUNCH INTERRUPT VECTOR.
2692  012016  012050                        CT12C
2693  012020  005067  166752        CT12A:  CLR     PSW             ;SET PRTY TO 0.
2694  012024  004767  174154                JSR     %7,CPRDY        ;CHECK FOR PUNCH READY.
2695  012030  042777  000100  170156        BIC     #BIT6,@PPS      ;DISABLE PUNCH INTERRUPT
2696  012036  052777  000100  170150        BIS     #BIT6,@PPS      ;ENABLE PUNCH INTERRUPT
2697  012044  000240                        NOP
2698  012046  104006              CT12E:    ERROR                   ;ERROR, FAILURE TO INTERRUPT WITH
2699                                                                 ;PUNCH READY BIT SET.
2700  012050  104013              CT12C:    SCOPE                   ;INTERRUPT VECTOR POINTS HERE.
2701                               ;*****************************************************************************
2702  012052  000013              CT13:     13                      ;TEST #
2703  012054  012144                        CT14                    ;NEXT TEST
2704  012056  001750                        1000.                   ;I COUNT
2705  012060  012062                        CT13A                   ;SCOPE ENTRY
2706                               ;*****************************************************************************
2707                               ;TEST THAT PUNCH DOES NOT REINTERRUPT AFTER RTI WHEN READY BIT IS NOT RESET.
2708  012062  104012              CT13A:    STPCHV                  ;SET UP PUNCH INTERRUPT VECTOR
2709  012064  012120                        CT13C
2710  012066  005067  166704                CLR     PSW             ;SET PRTY TO 0.
2711  012072  004767  174106                JSR     %7,CPRDY        ;CHECK FOR PUNCH READY.
2712  012076  042777  000100  170110        BIC     #BIT6,@PPS      ;DISABLE PUNCH INTERRUPT
2713  012104  052777  000100  170102        BIS     #BIT6,@PPS      ;ENABLE PUNCH INTERRUPT
2714  012112  000240                        NOP
2715  012114  104006              CT13E1:   ERROR                   ;ERROR1, PUNCH FAILED TO INTERRUPT.
2716  012116  104013                        SCOPE
2717  012120  012777  012140  170076  CT13C:  MOV     #CT13E2,@PCHVTR ;CHANGE INTERRUPT VECTOR TO CT13E2
2718  012126  012716  012134                MOV     #CT13D,@%6      ;CHANGE INTERRUPT RETURN ADDRESS.
2719  012132  000002                        RTI                     ;RETURN FROM INTERRUPT.
2720  012134  000240              CT13D:    NOP
2721  012136  104013                        SCOPE
2722  012140  104006              CT13E2:   ERROR                   ;ERROR2, PUNCH REINTERRUPTED AFTER RTI WITH
2723  012142  104013                        SCOPE                   ;READY BIT LEFT ON.
2724                               ;*****************************************************************************
2725  012144  000014              CT14:     14                      ;TEST #
2726  012146  012220                        CT15                    ;NEXT TEST
2727  012150  001750                        1000.                   ;I COUNT
2728  012152  012160                        CT14A                   ;SCOPE ENTRY
2729                               ;*****************************************************************************
2730                               ;TEST THAT THE PUNCH DOES NOT INTERRUPT WITH PROCESSOR AT SAME PRIORITY
2731                               ;LEVEL AS THE PUNCH.
2732  012154  104012                        STPCHV                  ;SET UP PUNCH INTERRUPT VECTOR.
```

```
2733  011156  011214                          CT14E
2734  011160  016767  170042  166610  CT14A:  MOV     PCHLVL,PSW      ;SET PROCESSOR PRIORITY SAME AS PUNCH.
2735  011166  005077  170022                  CLR     @PPS            ;DISABLE PUNCH INTERRUPT.
2736  011172  004767  174006                  JSR     %7,CPRDY        ;CHECK FOR PUNCH READY.
2737  011176  052777  000100  170010          BIS     #BIT6,@PPS      ;ENABLE PUNCH INTERRUPT.
2738  011204  000240                          NOP                     ;OR IF NO INTERRUPT OCCURS.
2739  011206  005077  170002                  CLR     @PPS            ;DISABLE PUNCH INTERRUPT.
2740  011212  104013                          SCOPE
2741  011214  104006                  CT14E:  ERROR                   ;ERROR, PUNCH ERRONEOUSLY INTERRUPTED
2742                                                                   ;WITH PROCESSOR AT SAME PRIORITY LEVEL
2743                                                                   ;AS THE PUNCH, OR THE PUNCH IS AT HIGHER
2744  011216  104013                          SCOPE                   ;PRIORITY LEVEL THAN SPECIFIED AT PCHLVL.
2745                                  ;*****************************************************************
2746  011220  000015                  CT15:   15                      ;TEST #
2747  011222  011276                          CT16                    ;NEXT TEST
2748  011224  001750                          1000.                   ;I COUNT
2749  011226  011234                          CT15A                   ;SCOPE ENTRY
2750                                  ;*****************************************************************
2751                                  ;TEST THAT PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY 1 LEVEL LOWER
2752                                  ;THAN THE PUNCH PRIORITY.
2753  011230  104012                          STPCHV                  ;SET UP PUNCH INTERRUPT VECTOR
2754  011232  011274                          CT15B
2755  011234  016767  167766  166534  CT15A:  MOV     PCHLVL,PSW      ;SET PROCESSOR PRIORITY ONE LEVEL LOWER
2756  011242  162767  000040  166526          SUB     #40,PSW         ;THAN PUNCH PRIORITY.
2757  011250  042777  000100  167736          BIC     #BIT6,@PPS      ;DISABLE PUNCH INTERRUPT
2758  011256  004767  173722                  JSR     %7,CPRDY        ;CHECK FOR PUNCH READY.
2759  011262  052777  000100  167724          BIS     #BIT6,@PPS      ;ENABLE PUNCH INTERRUPT.
2760  011270  000240                          NOP
2761  011272  104006                  CT15E:  ERROR                   ;PUNCH FAILED TO INTERRUPT WITH PROCESSOR
2762                                                                   ;PRIORITY ONE LEVEL LOWER THAN PUNCH.
2763                                                                   ;THEREFORE, PUNCH PRIORITY MUST
2764                                                                   ;BE LOWER THAN SPECIFIED AT PCHLVL.
2765  011274  104013                  CT15B:  SCOPE                   ;HERE IF INTERRUPT OCCURS.
2766                                  ;*****************************************************************
2767  011276  000016                  CT16:   16                      ;TEST #
2768  011300  011356                          CT17                    ;NEXT TEST
2769  011302  001750                          1000.                   ;I COUNT
2770  011304  011312                          CT16A                   ;SCOPE ENTRY
2771                                  ;*****************************************************************
2772                                  ;TEST THAT PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING CP PRIORITY TO 0.
2773  011306  104012                          STPCHV                  ;SET UP PUNCH INTERRUPT VECTOR
2774  011310  011354                          CT16B
2775  011312  012767  000340  166456  CT16A:  MOV     #PRTY7,PSW      ;SET PROCESSOR PRIORITY TO 7
2776  011320  004767  173660                  JSR     %7,CPRDY        ;CHECK FOR PUNCH READY.
2777  011324  042777  000100  167662          BIC     #BIT6,@PPS      ;DISABLE PUNCH INTERRUPT
2778  011332  052777  000100  167654          BIS     #BIT6,@PPS      ;ENABLE PUNCH INTERRUPT
2779  011340  005067  166432                  CLR     PSW             ;LOWER PRTY TO 0.
2780  011344  012767  000340  166424          MOV     #PRTY7,PSW      ;RAISE CP PRIORITY BACK TO 7.
2781  011352  104006                  CT16E:  ERROR                   ;ERROR, PUNCH FAILED TO INTERRUPT IMMEDIATELY
2782                                                                   ;AFTER CP PRIORITY WAS LOWERED TO 0.
2783  011354  104013                  CT16B:  SCOPE                   ;HERE IF INTERRUPT OCCURS.
2784                                  ;*****************************************************************
2785  011356  100017                  CT17:   17+MANUAL               ;TEST #
2786  011360  011504                          CT20                    ;NEXT TEST.
2787  011362  000144                          100.                    ;I COUNT
2788  011364  011410                          CT17A                   ;SCOPE ENTRY
```

```
2789                                    ;**********************************************************************
2790                                    ;TEST THAT THE PUNCH ERROR BIT IS ABLE TO INTERRUPT, AND THAT IT DOES NOT
2791                                    ;REINTERRUPT AFTER RTI.
2792  011366  004567  171162                       JSR     %5,PCSIM
2793  011372  000433                               433                     ;TURN PUN ERROR ON IF ON XOR TEST&R.
2794  011374  011410                               CT17A
2795  011376  104004                               TYPE8                   ;TYPE INSTRUCTION TO REMOVE TAPE FROM PUNCH
2796  011400  016567                               IM15
2797  011402  017025                               IM23
2798  011404  177777                               -1
2799  011406  000000                               HALT
2800  011410  104002                  CT17A:       SRESET                  ;RESET
2801  011412  104012                               STPCHV                  ;SET UP PUNCH INTERRUPT VECTOR.
2802  011414  011454                               CT17B
2803  011416  005777  167572                       TST     @PPS            ;TEST PPS
2804  011422  100026                               BPL     CT17E3          ;BRANCH IF ERROR BIT NOT SET.
2805  011424  112777  000000  167564               MOVB    #0,@PPB         ;0 TO PPB TO RESET READY.
2806  011432  042777  000100  167554               BIC     #BIT6,@PPS      ;DISABLE PUNCH INTERRUPT
2807  011440  052777  000100  167546               BIS     #BIT6,@PPS      ;ENABLE PUNCH INTERRUPT
2808  011446  000240                               NOP
2809  011450  104006                  CT17E1:      ERROR                   ;ERROR1,PUNCH ERROR BIT FAILED TO
2810  011452  104013                               SCOPE                   ;CAUSE INTERRUPT.
2811  011454  012777  011474  167542  CT17B:       MOV     #CT17E2,@PCHVTR ;CHANGE INTERRUPT VECTOR TO CT17E2
2812  011462  012716  011470                       MOV     #CT17C,@%6      ;CHANGE INTERRUPT RETURN ADDR TO CT17C
2813  011466  000002                               RTI                     ;RETURN FROM INTERRUPT
2814  011470  000240                  CT17C:       NOP                     ;HERE IF NO REINTERRUPT OCCURS.
2815  011472  104013                               SCOPE
2816  011474  104006                  CT17E2:      ERROR                   ;ERROR2, PUNCH REINTERRUPTED AFTER
2817  011476  104013                               SCOPE                   ;RTI. (ERROR BIT LEFT ON).
2818  011500  104006                  CT17E3:      ERROR                   ;ERROR3, ERROR BIT NOT SET.
2819  011502  104013                               SCOPE
2820                                    ;**********************************************************************
2821  011504  100020                  CT20:        20+MANUAL               ;TEST 8
2822  011506  177777                               -1                      ;LAST TEST
2823  011510  001750                               1000.                   ;I COUNT
2824  011512  011536                               CT20A                   ;SCOPE ENTRY
2825                                    ;**********************************************************************
2826                                    ;TEST THAT WITH ERROR BIT SET AND HAVING GENERATED AN INTERRUPT,
2827                                    ;LOADING THE PUNCH BUFFER CAUSES AN IMMEDIATE INTERRUPT.
2828  011514  004567  171034                       JSR     %5,PCSIM
2829  011520  000433                               433
2830  011522  011536                               CT20A
2831  011524  104004                               TYPE8                   ;MESSAGE TO REMOVE TAPE FROM PUNCH
2832  011526  016567                               IM15
2833  011530  017025                               IM23
2834  011532  177777                               -1
2835  011534  000000                               HALT
2836  011536  104002                  CT20A:       SRESET                  ;RESET.
2837  011540  104012                               STPCHV                  ;SET PTPI VECTOR TO CT20B.
2838  011542  011572                               CT20B
2839  011544  005777  167444                       TST     @PPS            ;TEST FOR ERROR
2840  011550  100025                               BPL     CT20E1          ;BRANCH IF ERROR BIT NOT SET.
2841  011552  005077  167436                       CLR     @PPS            ;DISABLE PTPI
2842  011556  052777  000100  167430               BIS     #BIT6,@PPS      ;ENABLE PTPI
2843  011564  000240                               NOP
2844  011566  104006                  CT20E2:      ERROR                   ;ERROR FAILED TO INTERRUPT.
```

```
 2845  011570  104013                          SCOPE
 2846  011572  012716  011600          CT20B:  MOV     #CT20C,@#6      ;ERROR INTERRUPT COMES HERE. SET UP
 2847  011576  000002                          RTI                     ;INTERRUPT EXIT TO CT20 AND EXIT.
 2848  011600  104012                  CT20C:  STPCHV                  ;SET PTPI VECTOR TO CT20D.
 2849  011602  011622                          CT20D
 2850  011604  005777  167404                  TST     @PPS            ;TEST ERROR
 2851  011610  100005                          BPL     CT20E1          ;BRANCH IF ERROR BIT NOT SET.
 2852  011612  005077  167400                  CLR     @PPB            ;LOAD PUNCH BUFFER.
 2853  011616  000240                          NOP
 2854  011620  104006                  CT20E3: ERROR                   ;BUFFER LOAD WITH PREVIOUS ERROR
 2855                                                                  ;INTERRUPT FAILED TO INTERRUPT.
 2856  011622  104013                  CT20D:  SCOPE                   ;OK, INTERRUPT OCCURRED.
 2857  011624  005077  167364          CT20E1: CLR     @PPS            ;CLEAR PTPI
 2858  011630  104006                          ERROR                   ;ERROR, ERROR BIT NOT SET.
 2859  011632  104013                          SCOPE
```

```
                                                        .SBTTL  PRG3 - PUNCH TEST
011634  012767  011662  167410  PRG3:   MOV     #DT0,KSTART     ;ADDR OF 1ST ROUTINE TO KSTART.
011642  104003                          TYPE                    ;TYPE TITLE.
011644  015627                          IMSB
011646  004767  003370                  JSR     %7,SWTL
011652  004767  172312                  JSR     PC,PTMCAL       ;CALIBRATE DELAY RTN WITH PUNCH.
011656  000167  170026                  JMP     GETRDY          ;GO GET STARTED
                                ;*******************************************************************
011662  000000          DT0:    0                               ;TEST #
011664  011732                  DT1                             ;NEXT TEST
011666  000005                  5                               ;I COUNT
011670  011672                  DT0A                            ;SCOPE ENTRY
                                ;*******************************************************************
                                ;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 0 (FULL SPEED)
011672  012767  001000  167476  DT0A:   MOV     #512.,RCNT      ;SET CHARACTER COUNT TO 512
011700  004567  000322                  JSR     %5,PFRNT        ;GO PUNCH FRONT END AND MODE 0
011704  000000                          0                       ;INDICATOR
011706  004767  172742                  JSR     %7,INBIN        ;INITIALIZE SPECIAL BINARY COUNT
011712  004767  172774          DT0B:   JSR     %7,GTBIN        ;GET BINARY CHARACTER
011716  004767  173314                  JSR     %7,MSPCH        ;GO PUNCH THE CHARACTER
011722  005367  167450                  DEC     RCNT            ;DECREMENT CHAR COUNT.
011726  001371                          BNE     DT0B            ;BRANCH IF COUNT NOT YET 0 YET.
011730  104013                          SCOPE
                                ;*******************************************************************
011732  000001          DT1:    1                               ;TEST #
011734  012012                  DT2                             ;NEXT TEST
011736  000005                  5                               ;I COUNT
011740  011750                  DT1A                            ;SCOPE ENTRY
                                ;*******************************************************************
                                ;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 1 (RANDOM STALLS AFTER
                                ;PUNCHING EACH CHARACTER. MAXIMUM STALL 47 MILLISECONDS)
011742  012767  177720  172336          MOV     #177720,STLMSK  ;SET STALL MASK FOR 57(8) MAX
011750  012767  001000  167420  DT1A:   MOV     #512.,RCNT      ;SET CHARACTER COUNT TO 512.
011756  004567  000244                  JSR     %5,PFRNT        ;GO PUNCH FRONT END, AND MODE 1
011762  000001                          1                       ;INDICATOR
011764  004767  172664                  JSR     %7,INBIN        ;INITIALIZE SPECIAL BINARY COUNT.
011770  004767  172716          DT1B:   JSR     %7,GTBIN        ;GET BINARY CHARACTER.
011774  004767  173236                  JSR     %7,MSPCH        ;GO PUNCH THE CHARACTER.
012000  104005                          STALL                   ;RANDOM STALL.
012002  005367  167370                  DEC     RCNT            ;DECREMENT CHAR COUNT.
012006  001370                          BNE     DT1B            ;BRANCH IF COUNT NOT YET 0.
012010  104013                          SCOPE
                                ;*******************************************************************
012012  000002          DT2:    2                               ;TEST #
012014  012114                  DT3                             ;NEXT TEST
012016  000005                  5                               ;I COUNT
012020  012036                  DT2A                            ;SCOPE ENTRY
                                ;*******************************************************************
                                ;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)
012022  012767  177720  172256          MOV     #177720,STLMSK  ;SET STALL MASK FOR 57(8) MAX.
012030  012767  177760  172330          MOV     #177760,RCMSK   ;SET CHAR GROUP MASK FOR 17(8) MAX).
012036  012767  001000  167332  DT2A:   MOV     #512.,RCNT      ;SET CHARACTER COUNT TO 512.
```

```
2916  012044  004567  000156            JSR     %5,PFRNT        ;GO PUNCH FRONT END AND MODE 2
2917  012050  000002                    2                       ;INDICATOR
2918  012052  004767  172576            JSR     %7,INBIN        ;INITIALIZE SPECIAL BINARY COUNT.
2919  012056  004767  172264  DT2B:     JSR     %7,GRCNT        ;GENERATE RANDOM CHARACTER COUNT
2920  012062  104005                    STALL                   ;RANDOM STALL.
2921  012064  004767  172622  DT2C:     JSR     %7,GTBIN        ;GET BINARY CHARACTER.
2922  012070  004767  173142            JSR     %7,HSPCH        ;PUNCH THE CHARACTER.
2923  012074  005367  167276            DEC     RCNT            ;DECREMENT CHAR COUNT
2924  012100  001404                    BEQ     DT2D            ;BRANCH IF COUNT IS 0.
2925  012102  005367  172262            DEC     RNCNT           ;NOT 0, DECREMENT RANDOM CHAR COUNT.
2926  012106  001366                    BNE     DT2C            ;BRANCH IF COUNT NOT YET 0.
2927  012110  000762                    BR      DT2B            ;BRANCH IF COUNT 0.
2928  012112  104013  DT2D:     SCOPE
2929                  ;*****************************************************************************
2930  012114  000003  DT3:     3                               ;TEST 0.
2931  012116  012206                    DT4                     ;NEXT TEST.
2932  012120  000001                    1                       ;I COUNT.
2933  012122  012124                    DT3A                    ;SCOPE ENTRY.
2934                  ;*****************************************************************************
2935                  ;PUNCH SPECIAL BINARY COUNT PATTERN IN PUNCH MODE 3.
2936                  ;STALL 5 SECONDS, PUNCH 32 CHARACTERS, UNTIL THE ENTIRE PATTERN IS
2937                  ;COMPLETED.
2938  012124  012767  001000  167244  DT3A:    MOV     #512.,RCNT      ;SET CHARACTER COUNT TO 512.
2939  012132  004567  000070            JSR     %5,PFRNT        ;GO PUNCH FRONT END AND MODE 3
2940  012136  000003                    3                       ;INDICATOR.
2941  012140  004767  172510            JSR     %7,INBIN        ;INITIALIZE SPECIAL BIN COUNT
2942  012144  104000  DT3B:     DELAY                           ;STALL 5 SECONDS
2943  012146  011610                    5000.
2944  012150  012767  000040  172212            MOV     #32.,RNCNT      ;SET GROUP COUNT TO 32.
2945  012156  004767  172530  DT3C:     JSR     %7,GTBIN        ;GET BINARY CHARACTER
2946  012162  004767  173050            JSR     %7,HSPCH        ;PUNCH CHARACTER
2947  012166  005367  167204            DEC     RCNT            ;DECREMENT CHAR COUNT
2948  012172  001404                    BEQ     DT3D            ;BRANCH IF COUNT IS 0
2949  012174  005367  172170            DEC     RNCNT           ;DECREMENT GROUP COUNT
2950  012200  001366                    BNE     DT3C            ;BRANCH IF COUNT NOT YET 0.
2951  012202  000760                    BR      DT3B            ;BRANCH IF COUNT 0.
2952  012204  104013  DT3D:     SCOPE
2953                  ;*****************************************************************************
2954  012206  000004  DT4:     4                               ;TEST#.
2955  012210  177777                    -1                      ;LAST TEST
2956  012212  000001                    1                       ;I COUNT.
2957  012214  012216                    DT4A                    ;SCOPE ENTRY POINT.
2958                  ;*****************************************************************************
2959                  ;THIS ROUTINE PROVIDES END OF PASS HALT FOR PRG3.
2960  012216  104003  DT4A:    TYPE                            ;TYPE END OF PASS
2961  012220  020201                    P3END
2962  012222  104010                    CHALT                   ;COMMON HALT.
2963  012224  104013                    SCOPE
2964                  ;SUBROUTINE TO PUNCH FRONT END AND MODE CODE (USED BY PRG3).
2965  012226  012701  000024  PFRNT:    MOV     #20.,%1         ;PUNCH 20 BLANK CHARACTERS (000)
2966  012232  005000                    CLR     %0              ;CLEAR R0
2967  012234  004767  172776            JSR     %7,HSPCH        ;PUNCH CHAR.
2968  012240  005301                    DEC     %1              ;DECREMENT R1
2969  012242  001374                    BNE     .-6             ;BRANCH IF NOT 20 CHARCTERS YET.
2970  012244  012700  000377            MOV     #377,%0         ;PUNCH RUBOUT CHAR (SYNC CHAR).
2971  012250  004767  172762            JSR     %7,HSPCH
```

```
 2972  012254  012500                    MOV     (5)+,R0         ;MOVE MODE CODE TO R0
 2973  012256  004767  172754             JSR     R7,HSPCH        ;PUNCH MODE CODE.
 2974  012262  012701  000004             MOV     #4,R1           ;PUNCH 4 BLANK CHARACTERS.
 2975  012266  005000                    CLR     R0
 2976  012270  004767  172742             JSR     R7,HSPCH
 2977  012274  005301                    DEC     R1
 2978  012276  001374                    BNE     .-6
 2979  012300  000205                    RTS     R5              ;EXIT.
 2980
 2981
```

```
 2982                                           .SBTTL  PRG4 - PUNCH VERIFY PROGRAM
 2983                                   ;THIS PROGRAM VERIFIES TAPE PRODUCED BY PRG3,
 2984                                   ;ANY ERRORS FOUND ARE REPORTED,
 2985  012302  104004                  PRG4:   TYPES                   ;TYPE TITLE AND INSTRUCTIONS
 2986  012304  016700                          IM20
 2987  012306  016105                          IM48
 2988  012310  016334                          IM6
 2989  012312  017025                          IM23
 2990  012314  177777                          -1
 2991  012316  000000                          HALT
 2992  012320  004767  002716                  JSR     %7,SWTL
 2993  012324  004767  171370                  JSR     PC,RTMCAL       ;CALIBRATE DELAY RTN WITH READER,
 2994  012330  012767  000372  167062  ET0A:   MOV     #250.,CTRA      ;250 TO CTRA,(TOTAL CHAR COUNT),
 2995  012336  012767  000012  167056  ET0B:   MOV     #10.,CTRB
 2996  012344  004767  170532          ET0C:   JSR     %7,BREAD        ;READ CHAR
 2997  012350  005767  167024                  TST     CRBUF
 2998  012354  001007                          BNE     ET0D            ;BRANCH IF NON-ZERO CHAR,
 2999  012356  005367  167040                  DEC     CTRB            ;0 CHAR, DECREMENT CTRB
 3000  012362  001412                          BEQ     ET0F            ;BRANCH IF 10 CONSECUTIVE 0'S READ,
 3001  012364  005367  167030                  DEC     CTRA            ;NO, DECREMENT CTRA,
 3002  012370  001365                          BNE     ET0C            ;BRANCH IF NOT YET 250 CHARS READ,
 3003  012372  000403                          BR      ET0E            ;250 CHARS READ, SYNC ERROR,
 3004  012374  005367  167020          ET0D:   DEC     CTRA            ;DECREMENT CTRA
 3005  012400  001356                          BNE     ET0B            ;BRANCH IF NOT 250 CHARS READ YET,
 3006  012402  104007                  ET0E:   ERROR1                  ;SYNC ERROR, 250 CHARS READ WITHOUT
 3007  012404  017747                          EM3                     ;A SUCCESSFUL SYNC,
 3008  012406  000750                          BR      ET0A            ;GO TRY AGAIN,
 3009  012410  004767  170466          ET0F:   JSR     %7,BREAD        ;READ CHAR
 3010  012414  005767  166760                  TST     CRBUF
 3011  012420  001004                          BNE     ET0G            ;BRANCH IF NON-ZERO CHAR,
 3012  012422  005367  166772                  DEC     CTRA            ;DECREMENT CTRA
 3013  012426  001370                          BNE     ET0F            ;BRANCH IF NOT 250 CHARS READ YET,
 3014  012430  000764                          BR      ET0E            ;250 CHARS READ, SYNC ERROR,
 3015  012432  022767  000377  166740  ET0G:   CMP     #377,CRBUF      ;COMPARE CHAR READ TO 377,
 3016  012440  001416                          BEQ     ET0H            ;377,OK,
 3017  012442  012767  000377  166712          MOV     #377,ERRT       ;NOT 377,LEADER ERROR, SET UP FOR
 3018  012450  004567  172400                  JSR     %5,ACNV4        ;ERROR TYPEOUT,
 3019  012454  001362                          ERRT
 3020  012456  020011                          ESB
 3021  012460  004567  172370                  JSR     %5,ACNV4
 3022  012464  001400                          CRBUF
 3023  012466  020024                          EWAS
 3024  012470  104007                          ERROR1                  ;LEADER ERROR, SHOULD BE 377,
 3025  012472  017764                          EM4
 3026  012474  000715                          BR      ET0A            ;START OVER
 3027  012476  004767  170400          ET0H:   JSR     %7,BREAD        ;READ CHAR,
 3028  012502  026727  166672  000003          CMP     CRBUF,#3        ;COMPARE CHAR READ TO 3,
 3029  012510  101407                          BLOS    ET0I            ;BRANCH IF SAME OR LOWER,
 3030  012512  004567  172336                  JSR     %5,ACNV4        ;ERROR, CONVERT DATA READ TO ASCII,
 3031  012516  001400                          CRBUF                   ;SET UP FOR TYPEOUT,
 3032  012520  020103                          FWAS
 3033  012522  104007                          ERROR1                  ;LEADER ERROR, SHOULD BE BETWEEN
 3034  012524  020031                          EM5                     ;0 AND 3,
 3035  012526  000700                          BR      ET0A            ;START OVER,
 3036  012530  012767  000004  166662  ET6I:   MOV     #4,CTRA         ;4 TO CTRA (CHAR COUNT)
 3037  012536  005067  166620                  CLR     ERRT            ;CLEAR ERRT, EXPECTED CHAR IS 0,
```

```
3038  012542  004767  170334          ET0J:  JSR     %7,BREAD        ;READ CHAR,
3039  012546  004767  000050                 JSR     %7,ECHK         ;GO CHECK CHAR READ,
3040  012552  005367  166642                 DEC     CTRA            ;DECREMENT CTRA
3041  012556  001371                        BNE     ET0J            ;BRANCH IF NOT 4 CHARS READ YET,
3042  012560  004767  172070                 JSR     %7,INBIN        ;INITIALIZE SPECIAL BINARY COUNT,
3043  012564  012767  001000  166626         MOV     #512.,CTRA      ;SET CHAR COUNT TO 512,
3044  012572  004767  170304          ET0K:  JSR     %7,BREAD        ;READ CHAR,
3045  012576  004767  172110                 JSR     %7,GTBIN        ;GET BIN CHAR AND STORE AT
3046  012602  010067  166554                 MOV     %0,ERRT         ;ERRT(HOLDS EXPECTED DATA),
3047  012606  004767  000010                 JSR     %7,ECHK         ;GO CHECK CHAR READ,
3048  012612  005367  166602                 DEC     CTRA            ;DECREMENT CHAR COUNT
3049  012616  001365                        BNE     ET0K            ;BRANCH IF NOT 512 CHARS READ YET,
3050  012620  000643                        BR      ET0A            ;DONE, START OVER,
3051  012622  026767  166552  166532  ECHK:  CMP     CRBUF,ERRT      ;COMPARE CHAR READ AGAINST EXPECTED CHAR,
3052  012630  001412                        BEQ     ECHKA           ;BRANCH IF EQUAL,
3053  012632  004567  172216                 JSR     %5,ACNV4        ;CONVERT EXPECTED DATA TO ASCII,
3054  012636  001362                        ERRT
3055  012640  017655                        ASB
3056  012642  004567  172206                 JSR     %5,ACNV4        ;CONVERT DATA READ TO ASCII,
3057  012646  001400                        CRBUF
3058  012650  017670                        AWAB
3059  012652  104007                        ERROR1                  ;ERROR, DATA ERROR,
3060  012654  017632                        EM1
3061  012656  000207          ECHKA: RTS     %7              ;EXIT
```

```
                                        .SBTTL  PRGS THROUGH  PRG13
                                ;*****************************************************************
                                ;PRGS - COMBINED READER PUNCH TEST, USES SPECIAL
                                ;*****************************************************************
                                ;BINARY COUNT PATTERN.
012660  104004          PRGS:   TYPES                   ;TYPE TITLE AND INSTRUCTIONS.
012662  017252                  IM26
012664  016334                  IM6
012666  017025                  IM23
012670  177777                  -1
012672  000000                  HALT
012674  004767  002342          JSR     %7,SNTL
012700  004767  171264          JSR     PC,PTMCAL       ;CALIBRATE DELAY RTN WITH PUNCH.
012704  004767  171744          JSR     %7,INBIN        ;INITIALIZE BINARY COUNTS.
012710  012767  177600  171370  MOV     #177600,STLMSK  ;SET MAX. STALL DELAY.
012716  005067  000312          CLR     PCHCNT          ;CLEAR PUNCH COUNT
012722  005067  000310          CLR     RBUSY           ;CLEAR READER BUSY INDICATOR
012726  104011                  STRDRV                  ;SET PTRI VECTOR.
012730  013240                  WNZERO
012732  104012                  STPCHV                  ;SET PTPI VECTOR.
012734  012766                  PBIN
012736  004767  167700          JSR     %7,ARRDY        ;CHECK FOR READER READY
012742  004767  172236          JSR     %7,CPRDY        ;CHECK FOR PUNCH READY
012746  004767  172006          JSR     %7,GTBINP       ;GET BIN CHARACTER
012752  010177  166240          MOV     %1,@PPB         ;PUNCH IT
012756  052777  000100  166230  BIS     #BIT6,@PPS      ;ENABLE PTPI
012764  000777                  BR      .
012766  005777  166222  PBIN:   TST     @PPS            ;TEST FOR ERROR.
012772  100004                  BPL     PBNA            ;BRANCH IF NO ERROR.
012774  104003                  TYPE                    ;TYPE PUNCH NOT READY
012776  017456                  SM3                     ;MESSAGE.
013000  104010                  CHALT
013002  000771                  BR      PBIN            ;RECHECK FOR ERROR.
013004  105777  166204  PBNA:   TSTB    @PPS            ;CHECK FOR DONE.
013010  100402                  BMI     PBNB            ;BRANCH IF DONE SET.
013012  104007                  ERROR1                  ;ERROR.FALSE PUNCH INTERRUPT.
013014  020161                  EM11
013016  005267  000212  PBNB:   INC     PCHCNT          ;INCREMENT PUNCH COUNT.
013022  004767  171732          JSR     %7,GTBINP       ;GET BINARY CHARACTER
013026  010177  166164          MOV     %1,@PPB         ;ENABLE PUNCH
013032  105767  000200          TSTB    RBUSY           ;CHECK READER BUSY INDICATOR
013036  100414                  BMI     PBINA           ;BRANCH IF READER BUSY
013040  026727  000170  000024  CMP     PCHCNT,#20.     ;NOT BUSY, PUNCH COUNT 20 YET?
013046  103001                  BHIS    .+4             ;BRANCH IF PCHNT 20 OR MORE.
013050  000002                  RTI                     ;NOT 20 YET, EXIT INTERRUPT
013052  052767  000200  000156  BIS     #BIT7,RBUSY     ;SET READER BUSY
013060  052777  000101  166122  BIS     #101,@PRS       ;ENABLE PTRI AND READER.
013066  000002                  RTI                     ;EXIT INTERRUPT.
013070  026727  000140  000050  PBINA:  CMP     PCHCNT,#40.     ;PUNCH COUNT LARGER THAN 40?
```

```
3111  013076  101402                          BLOS    PBINB           ;BRANCH IF NOT LARGER
3112  013100  005077  166110                   CLR     @PPS            ;LARGER, DISABLE PTPI
3113  013104  000002                  PBINB:  RTI                     ;EXIT INTERRUPT.
3114  013106  005777  166076          CREAD:  TST     @PRS            ;CHECK FOR ERROR.
3115  013112  100003                          BPL     CRDA            ;BRANCH IF NO ERROR.
3116  013114  004767  167576                   JSR     %7,TSM2         ;ERROR,TYPE MESSAGE.
3117  013120  000772                          BR      CREAD
3118  013122  105777  166062          CRDA:   TSTB    @PRS            ;TEST FOR DONE.
3119  013126  100402                          BMI     CRDAA           ;BRANCH IF DONE SET.
3120  013130  104007                          ERROR1                  ;ERROR, FALSE READER INTERRUPT.
3121  013132  020140                          EM10
3122  013134  017767  166052  166236  CRDAA:  MOV     @PRB,CRBUF      ;CHARACTER READ TO CRBUF
3123  013142  005367  000066                  DEC     PCHCNT
3124  013146  026727  000062  000037          CMP     PCHCNT,#31.
3125  013154  101024                          BHI     CREADC          ;PUNCH COUNT GREATER THAN 31?
3126  013156  032777  000100  166030          BIT     #BIT6,@PPS      ;NO.
3127  013164  001003                          BNE     CREADA          ;PTPI ENABLED?
3128  013166  052777  000100  166020          BIS     #BIT6,@PPS      ;NO, ENABLE PTPI.
3129  013174  005767  000034          CREADA: TST     PCHCNT
3130  013200  001006                          BNE     CREADB          ;PUNCH COUNT 0?
3131  013202  042767  000200  000026          BIC     #BIT7,RBUSY     ;YES, CLEAR READER BUSY.
3132  013210  005077  165774                   CLR     @PRS            ;DISABLE PTRI.
3133  013214  000207                          RTS     %7              ;EXIT.
3134  013216  005767  000014          CREADB: TST     RBUSY           ;TEST BUSY INDICATOR
3135  013222  100401                          BMI     CREADC          ;STALL?
3136  013224  104005                          STALL                   ;YES.
3137  013226  005277  165756          CREADC: INC     @PRS            ;ENABLE READER
3138  013232  000207                          RTS     %7              ;EXIT.
3139  013234  000000                  PCHCNT: OPEN
3140  013236  000000                  RBUSY:  OPEN
3141  013240  004767  177642          NNZERO: JSR     %7,CREAD        ;READ CHARACTER
3142  013244  005767  166130                   TST     CRBUF           ;CHECK CHARACTER
3143  013250  001001                          BNE     .+4             ;BRANCH IF NON-ZERO CHAR.
3144  013252  000002                          RTI                     ;ZERO, EXIT INTERRUPT.
3145  013254  012777  013272  165736          MOV     #RBIN,@RDRVTR   ;SET READER VECTOR TO READ BINARY
3146  013262  012767  000003  166126          MOV     #3,ERCTR        ;COUNT. SET ERROR COUNTER TO 3.
3147  013270  000402                          BR      RBINA
3148  013272  004767  177610          RBIN:   JSR     %7,CREAD        ;READ CHARACTER.
3149  013276  004767  171410          RBINA:  JSR     %7,GTBIN        ;GET BINARY CHARACTER
3150  013302  020067  166072                   CMP     %0,CRBUF        ;COMPARE AGAINST CHAR READ.
3151  013306  001001                          BNE     RBINB           ;BRANCH IF NOT SAME.
3152  013310  000002                          RTI                     ;SAME EXIT INTERRUPT.
3153  013312  010067  166044          RBINB:  MOV     %0,ERRT         ;MOVE EXPECTED CHAR TO ERRT
3154  013316  004567  171532                   JSR     %5,ACNV4        ;CONVERT EXPECTED CHAR TO ASCII
3155  013322  001362                          ERRT
3156  013324  017655                          ASB
3157  013326  004567  171522                   JSR     %5,ACNV4        ;CONVERT RECEIVED CHAR TO ASCII
3158  013332  001400                          CRBUF
3159  013334  017670                          AWAS
3160  013336  104007                          ERROR1                  ;ERROR MESSAGE, DATA ERROR.
3161  013340  017632                          EM1
3162  013342  005367  166050                   DEC     ERCTR           ;3RD ERROR?
3163  013346  001001                          BNE     RBINC           ;YES.
3164  013350  000002                          RTI                     ;NO, EXIT INTERRUPT.
3165  013352  052767  100000  177656  RBINC:  BIS     #BIT15,RBUSY    ;DISABLE STALLS.
3166  013360  012777  013402  165632          MOV     #RBIND,@RDRVTR  ;SET PTR VECTOR TO RBIND.
```

```
3167  013366  012767  000003  166022          MOV     #3,ERCTR        ;USE ERCTR AS CHARACTER COUNTER.
3168  013374  012700  001402                  MOV     #CHR1,%0        ;ADDR OF CHR1 TO %0
3169  013400  000002                          RTI                     ;EXIT INTERRUPT
3170  013402  004767  177500          RBIND:  JSR     %7,CREAD        ;READ CHARACTER
3171  013406  016720  165766                  MOV     CRBUF,(0)+      ;STORE CHARACTER STARTING AT CHR1
3172  013412  005367  '56000                  DEC     ERCTR           ;3RD CHARACTER?
3173  013416  001401                          BEQ     .+4             ;YES.
3174  013420  000002                          RTI                     ;EXIT INTERRUPT. NOT 3RD YET.
3175  013422  0047    171'.4                  JSR     %7,SYNCA        ;GO SYNC THE READER.
3176  013426  000751                          BR      RBINC           ;NO SYNC. TRY AGAIN.
3177  013430  012777  013272  165562          MOV     #RBIN,@RDRVTR   ;SYNCED. SET READER VECTOR TO RBIN.
3178  013436  012767  000003  165752          MOV     #3,ERCTR        ;SET ERROR COUNT TO 3.
3179  013444  042767  100000  177564          BIC     #BIT15,RBUSY    ;ENABLE STALLS.
3180  013452  000002                          RTI                     ;EXIT INTERPUPT.
3181
3182
3183                                  ;*****************************************************************
3184                                  ;PRG6 - PUNCHES CONTINUOUSLY ON PTP THE 2 CHARACTERS SELECTED
3185                                  ;*****************************************************************
3186  013454  104004                  PRG6:   TYPES                   ;TYPE TITLE AND INSTRUCTIONS.
3187  013456  017003                          IM21
3188  013460  016620                          IM16
3189  013462  177777                          -1
3190  013464  012767  000004  165700  1$:     MOV     #4,COUNT
3191  013472  012767  020425  001224          MOV     #$CH1,TLX
3192  013500  104014                          OPTSEL
3193  013502  022767  000004  165662          CMP     #4,COUNT
3194  013510  001765                          BEQ     1$
3195  013512  116767  165646  000064          MOVB    TMP1,PUNC1
3196  013520  012767  000004  165644  2$:     MOV     #4,COUNT
3197  013526  012767  020463  001170          MOV     #$CH2,TLX
3198  013534  104014                          OPTSEL
3199  013536  022767  000004  165626          CMP     #4,COUNT
3200  013544  001765                          BEQ     2$
3201  013546  116767  165612  000031          MOVB    TMP1,PUNC1+1
3202  013554  104003                          TYPE
3203  013556  017025                          IM23
3204  013560  000000                          HALT
3205  013562  116700  000016          PRG6A:  MOVB    PUNC1,%0                ;PUNCH FIRST CHARACTER.
3206  013566  004767  171444                  JSR     %7,HSPCH
3207  013572  116700  000007                  MOVB    PUNC1+1,%0              ;PUNCH SECOND CHARACTER.
3208  013576  004767  171434                  JSR     %7,HSPCH
3209  013602  000767                          BR      PRG6A           ;REPEAT.
3210  013604  000000          PUNC1:.WORD 0
3211                                  ;*****************************************************************
3212                                  ;PRG7 - READS AND CHECKS TAPE PUNCHED WITH 2 CHARACTERS SELECTED
3213                                  ;*****************************************************************
3214  013606  104003                  PRG7:   TYPE
3215  013610  017014                          IM22
3216  013612  012767  000004  165552  2$:     MOV     #4,COUNT
3217  013620  012767  020617  001076          MOV     #$RD1,TLX
3218  013626  104014                          OPTSEL
3219  013630  022767  000004  165534          CMP     #4,COUNT
3220  013636  001765                          BEQ     2$
3221  013640  016767  165520  165556          MOV     TMP1,CTRC
3222  013646  012767  000004  165516  1$:     MOV     #4,COUNT
```

```
 3223  013654  012767  020654  001042          MOV     #SRD2,TLX
 3224  013662  104014                          OPTSEL
 3225  013664  022767  000004  165500          CMP     #4,COUNT
 3226  013672  001765                          BEQ     1$
 3227  013674  016767  165464  165524          MOV     TMP1,CTRD
 3228  013702  104004                          TYPES
 3229  013704  016334                          IM6
 3230  013706  017025                          IM23
 3231  013710  177777                          -1
 3232  013712  000000                          HALT
 3233  013714  004767  001322                  JSR     %7,SWTL
 3234  013720  004767  167156          HT0A:   JSR     %7,BREAD        ;MATCH CHARS ON TAPE AGAINST EXPECTED CHARS.
 3235  013724  016767  165450  165450          MOV     CRBUF,CHR1      ;READ CHAR INTO CHR1
 3236  013732  004767  167144                  JSR     %7,BREAD        ;READ CHAR
 3237  013736  016767  165436  165440          MOV     CRBUF,CHR2      ;INTO CHR2
 3238  013744  026767  165432  165452          CMP     CHR1,CTRC       ;(CHR1)=(CTRC)?
 3239  013752  001040                          BNE     HT0E            ;NO.
 3240  013754  026767  165424  165444          CMP     CHR2,CTRD       ;YES. (CHR2)=(CTRD)?
 3241  013762  001061                          BNE     HT0G            ;NO. MATCH ERROR.
 3242  013764  005067  165432                  CLR     CTRB            ;YES. NEXT CHAR SHOULD = (CTRC) (CTRB=0)
 3243  013770  012767  000003  165420  HT0B:   MOV     #3,ERCTR        ;3 TO ERROR COUNTER.
 3244  013776  004767  167100          HT0C:   JSR     %7,BREAD        ;READ CHAR
 3245  014002  005167  165414                  COM     CTRB            ;COMPLEMENT CHAR INDICATOR
 3246  014006  001436                          BEQ     HT0F            ;BRANCH IF EXPECTED CHAR SHOULD = (CTRD)
 3247  014010  026767  165364  165406          CMP     CRBUF,CTRC      ;(CRBUF) = (CTRC)?
 3248  014016  001767                          BEQ     HT0C            ;YES.
 3249  014020  004567  171030                  JSR     %5,ACNV4        ;NO. (CTRC) TO ASB IN ASCII FORM.
 3250  014024  001424                          CTRC
 3251  014026  017655                          ASB
 3252  014030  004567  171020          HT0D:   JSR     %5,ACNV4        ;(CRBUF) TO AWAS IN ASCII FORM.
 3253  014034  001400                          CRBUF
 3254  014036  017670                          AWAS
 3255  014040  104007                          ERROR1                  ;ERROR 1 CALL. TYPE DATA ERROR MESSAGE.
 3256  014042  017632                          EM1
 3257  014044  005367  165346                  DEC     ERCTR           ;3 ERRORS?
 3258  014050  001723                          BEQ     HT0A            ;YES. START ALL OVER.
 3259  014052  000751                          BR      HT0C            ;NO. CONTINUE READING.
 3260  014054  026767  165322  165344  HT0E:   CMP     CHR1,CTRD       ;(CHR1) = (CTRD)?
 3261  014062  001021                          BNE     HT0G            ;NO. MATCH ERROR.
 3262  014064  026767  165314  165332          CMP     CHR2,CTRC       ;YES. (CHR2) = (CTRC)?
 3263  014072  001015                          BNE     HT0G            ;NO. MATCH ERROR.
 3264  014074  012767  177777  165320          MOV     #-1,CTRB        ;YES. NEXT CHAR SHOULD = (CTRD)
 3265  014102  000732                          BR      HT0B            ;GO START READING.
 3266  014104  026767  165270  165314  HT0F:   CMP     CRBUF,CTRD      ;(CRBUF) = (CTRD)?
 3267  014112  001731                          BEQ     HT0C            ;YES. OK. CONTINUE READING.
 3268  014114  004567  170734                  JSR     %5,ACNV4        ;NO. (CTRD) TO ASB IN ASCII FORM.
 3269  014120  001426                          CTRD
 3270  014122  017655                          ASB
 3271  014124  000741                          BR      HT0D            ;GO GENERATE ERROR MESSAGE.
 3272  014126  104007                  HT0G:   ERROR1                  ;MATCH ERROR. UNABLE TO MATCH UP
 3273  014130  020110                          EM6                     ;2 CONSECUTIVE CHARACTERS FROM READER
 3274  014132  000672                          BR      HT0A            ;TO CHARACTERS READ FROM SR.
 3275                          ;*****************************************************************
 3276                          ;PRG10 - READ X CHARACTERS, STALL Y MILLISECONDS.
 3277                          ;*****************************************************************
 3278  014134  005067  165226  PRG10: CLR     TMP2
```

```
3279  014140  104003                          TYPE
3280  014142  015643                          IN17
3281  014144  012767  000004  165220  10:     MOV     #4,COUNT
3282  014152  012767  020375  000544          MOV     #SNUNCR,TLX
3283  014160  104014                          OPTSEL
3284  014162  022767  000004  165202          CMP     #4,COUNT
3285  014170  001765                          BEQ     10
3286  014172  116767  165166  165166          MOVB    TMP1,TMP2
3287  014200  012767  000004  165164  20:     MOV     #4,COUNT
3288  014206  012767  020357  000510          MOV     #STALL,TLX
3289  014214  104014                          OPTSEL
3290  014216  022767  000004  165146          CMP     #4,COUNT
3291  014224  001765                          BEQ     20
3292  014226  116767  165132  165133          MOVB    TMP1,TMP2+1
3293  014234  104003                          TYPE
3294  014236  017025                          IN23
3295  014240  000000                          HALT
3296  014242  004767  167452                  JSR     PC,RTMCAL       ;CALIBRATE DELAY RTN WITH READER.
3297  014246  005067  000042          ITA:    CLR     ITY
3298  014252  005067  000042                  CLR     ITX
3299  014256  116767  165105  000030          MOVB    TMP2+1,ITY              ;MOVE STALL COUNT TO ITY.
3300  014264  116767  165076  000026          MOVB    TMP2,ITX        ;MOVE CHAR COUNT TO ITX.
3301  014272  004767  166364          ITB:    JSR     %7,AREAD        ;FETCH CHARACTER.
3302  014276  105367  000016                  DECB    ITX             ;DECREMENT CHAR COUNT.
3303  014302  001373                          BNE     ITB             ;BRANCH IF COUNT NOT 0.
3304  014304  005767  000004                  TST     ITY             ;DELAY COUNT = 0?
3305  014310  001756                          BEQ     ITA             ;BR IF YES.
3306  014312  104000                          DELAY                   ;READ CHARS, STALL NOW.
3307  014314  000000          ITY:    OPEN                            ;STALL COUNT IN MSECS.
3308  014316  000753                          BR      ITA             ;REPEAT
3309  014320  000000          ITX:    OPEN
3310                                ;*************************************************************************
3311                                ;PRG11, PUNCH SPECIAL BINARY COUNT PATTERN TEST TAPE
3312                                ;*************************************************************************
3313  014322  104004          PRG11:  TYPES                           ;TYPE TITLE AND INSTRUCTIONS.
3314  014324  015653                  IN0C
3315  014326  016620                  IN16
3316  014330  017025                  IN23
3317  014332  177777                  -1
3318  014334  000000                  HALT                            ;WAIT FOR USER
3319  014336  012746  000024          MOV     #20.,-(6)               ;PUNCH 20 BLANK CHAR, LEADER
3320  014342  005000                  CLR     %0
3321  014344  004767  170666  PRG11A: JSR     %7,HSPCH
3322  014350  005316                  DEC     @%6
3323  014352  001374                  BNE     PRG11A
3324  014354  004767  170274          JSR     %7,INBIN                ;INITIALIZE SPECIAL BINARY COUNT
3325  014360  004767  170326  PRG11B: JSR     %7,GTBIN                ;GET BINARY CHARACTER.
3326  014364  004767  170646          JSR     %7,HSPCH                ;PUNCH CHARACTER
3327  014370  000773                  BR      PRG11B                  ;REPEAT.
3328                                ;*************************************************************************
3329                                ;PRG12 - READER SPEED PRINT LOOP
3330                                ;*************************************************************************
3331  014372  012767  000004  164772  PRG12:  MOV     #4,COUNT
3332  014400  012767  020340  000316          MOV     #STIME,TLX
3333  014406  104014                          OPTSEL
3334  014410  104003                          TYPE
```

```
3335  014412  020521                          $SKEY
3336  014414  000000                          HALT
3337  014416  005067  165002          KTA:    CLR     CTRC            ;CLEAR WORK REGISTERS
3338  014422  005067  164774                  CLR     CTRB
3339  014426  005077  164600                  CLR     @TKB
3340  014432  032767  000200  164724          BIT     #BIT7,TMP1
3341  014440  001403                          BEQ     KTB             ;300 SECOND TIMING IS DESIRED
3342  014442  012767  000416  164754          MOV     #270.,,CTRC     ;SET UP FOR DESIRED TIME BASE.
3343  014450  062767  000036  164746  KTB:    ADD     #30.,,CTRC
3344  014456  000407                          BR      KTD
3345  014460  004767  166176          KTC:    JSR     %7,AREAD        ;READ CHARACTER.
3346  014464  005367  164730                  DEC     CTRA            ;DECREMENT CTRA
3347  014470  001005                          BNE     KTE             ;BRANCH IF CTRA NOT 0.
3348  014472  005267  164724                  INC     CTRB            ;CTRA0.+1 TO CTRB.
3349  014476  016767  164722  164714  KTD:    MOV     CTRC,CTRA       ;RELOAD CTRA.
3350  014504  105777  164520          KTE:    TSTB    @TKS
3351  014510  100363                          BPL     KTC             ;NO.
3352  014512  004567  000100          KTF:    JSR     %5,CPKPL        ;GO TYPE OUT DEVICE SPEED.
3353  014516  017501                          $M4
3354  014520  000000                          HALT
3355  014522  000723                          BR      PRG12
3356                                  ;*****************************************************************
3357                                  ;PRG13 - PUNCH SPEED PRINT LOOP
3358                                  ;*****************************************************************
3359  014524  104004          PRG13:  TYPES                           ;TYPE TITLE AND INSTRUCTIONS.
3360  014526  017221                  IM25
3361  014530  016620                  IM16
3362  014532  020521                  $SKEY
3363  014534  177777                  -1
3364  014536  000000                  HALT                            ;HALT, WAIT FOR USER.
3365  014540  005067  164656  LTA:    CLR     CTRB            ;CLEAR WORK AREAS.
3366  014544  005000                  CLR     %0
3367  014546  005077  164460                  CLR     @TKB
3368  014552  000407                          BR      LTC
3369  014554  004767  170456  LTB:    JSR     %7,HSPCH        ;PUNCH A 0
3370  014560  005367  164634                  DEC     CTRA            ;DECREMENT CTRA
3371  014564  001005                          BNE     LTD             ;BRANCH IF CTRA NOT 0
3372  014566  005267  164630                  INC     CTRB            ;INCREMENT CTRB.
3373  014572  012767  000074  164620  LTC:    MOV     #60.,,CTRA      ;MOVE 60 TO CTRA
3374  014600  105777  164424          LTD:    TSTB    @TKS            ;TIME UP?
3375  014604  100363                          BPL     LTB             ;NO.
3376  014606  004567  000004          LTE:    JSR     %5,CPKPL        ;GO TYPE OUT DEVICE SPEED.
3377  014612  017523                          $M5
3378  014614  000750                          BR      LTA-2           ;GO HALT AND READY UP FOR NEXT TIME.
3379  014616  012567  000022          CPKPL:  MOV     (5)+,CPKPLA     ;MOVE ADDR OF 1ST MESSAGE TO CPKPLA.
3380  014622  004567  170430                  JSR     %5,BDCNV        ;CONVERT (CTRB) TO DECIMAL ASCII.
3381  014626  001422                          CTRB
3382  014630  004567  170332                  JSR     %5,BMOVE        ;MOVE 3 DECIMAL CHARS TO PRINTOUT AREA.
3383  014634  015260                          DECVAL+2
3384  014636  017544                          ACPS
3385  014640  000003                          3
3386  014642  104004                          TYPES                   ;TYPE DEVICE SPEED.
3387  014644  000000          CPKPLA: OPEN
3388  014646  017544                  ACPS
3389  014650  177777                  -1
3390  014652  000205                  RTS     %5              ;EXIT.
```

```
3391
3392
3393  014654  005077  164350          TTIN:   CLR     @TKB
3394  014660  005077  164346                  CLR     @TKB
3395  014664  005067  164504                  CLR     TIB
3396  014670  105777  164334          1$:     TSTB    @TKS
3397  014674  100375                          BPL     1$
3398  014676  017767  164330  164470          MOV     @TKB,TIB
3399  014704  105777  164324          2$:     TSTB    @TPS
3400  014710  100375                          BPL     2$
3401  014712  116777  164456  164316          MOVB    TIB,@TPB
3402  014720  000002                          RTI
3403
3404
3405  014722  104003                  OPTS:   TYPE
3406  014724  000000                  TLX:    OPEN
3407  014726  005067  164432                  CLR     TMP1
3408  014732  104016                  1$:     TTYIN
3409  014734  104017                          VALID
3410  014736  000775                          BR      1$
3411
3412
3413  014740  042767  177600  164426  VALINP: BIC     #177600,TIB
3414  014746  122767  000007  164420          CMPB    #7,TIB
3415  014754  001002                          BNE     11$
3416  014756  104015                          CNTL
3417  014760  000404                          BR      6$
3418  014762  122767  000025  164404  11$:    CMPB    #25,TIB
3419  014770  001004                          BNE     1$
3420  014772  022626                  6$:     POP&P2
3421  014774  162716  000016                  SUB     #16,(SP)
3422  015000  000002                          RTI
3423  015002  122767  000015  164364  1$:     CMPB    #15,TIB
3424  015010  001004                          BNE     4$
3425  015012  104003                          TYPE
3426  015014  020230                          $CRLF
3427  015016  022626                  9$:     POPSP2
3428  015020  000002                          RTI
3429  015022  122767  000012  164344  4$:     CMPB    #12,TIB
3430  015030  001410                          BEQ     5$
3431  015032  122767  000060  164334  2$:     CMPB    #60,TIB
3432  015040  003004                          BGT     5$
3433  015042  122767  000067  164324          CMPB    #67,TIB
3434  015050  002003                          BGE     7$
3435  015052  104003                  5$:     TYPE
3436  015054  020232                          $QUEST
3437  015056  000745                          BR      6$
3438  015060  006367  164300          7$:     ASL     TMP1
3439  015064  006367  164274                  ASL     TMP1
3440  015070  006367  164270                  ASL     TMP1
3441  015074  042767  177770  164272          BIC     #177770,TIB
3442  015102  056767  164266  164254          BIS     TIB,TMP1
3443  015110  005367  164256                  DEC     COUNT
3444  015114  001786                          BEQ     5$
3445  015116  000002                          RTI
3446
```

F-D EL

```
3467
3468  015120  105777  164104          CKSWRR: TSTB    @TKS
3469  015124  100046                          BPL     OUT
3470  015126  017767  164100  164240          MOV     @TKB,TIB
3471  015134  042767  177600  164232          BIC     #177600,TIB
3472  015142  022767  000007  164224          CMP     #7,TIB
3473  015150  001033                          BNE     OUT
3474  015152  104003                          TYPE
3475  015154  020223                          $CTLG
3476  015156  017767  164022  164200  CNTLU:  MOV     @SWR,TMP1
3477  015164  004567  167636                  JSR     %5,ACNV6
3478  015170  001364                          TMP1
3479  015172  020236                          $VALUE
3480  015174  104004                          TYPES
3481  015176  020304                          $SWREQ
3482  015200  020236                          $VALUE
3483  015202  177777                          -1
3484  015204  012767  020273  177512          MOV     @$NEW,TLX
3485  015212  012767  000007  164152          MOV     #7,COUNT
3486  015220  104014                          OPTSEL
3487  015222  022767  000007  164142          CMP     #7,COUNT
3488  015230  001403                          BEQ     OUT
3489  015232  016777  164126  163744          MOV     TMP1,@SWR
3490  015240  000002                  OUT:    RTI
3491
3492
3493  015242  104004                  SWTL:   TYPES
3494  015244  015363                          CM4
3495  015246  015453                          CM4B
3496  015250  177777                          -1
3497  015252  104015                          CNTL
3498  015254  000207                          RTS     %7
3499
```

```
3480                                                    .SBTTL
3481  015256  020040  020040     040  DECVAL: .ASCII  '     '
3482  015263     007                  APGEND: .BYTE   7
3483  015264  025045     100                .ASCII  '%=@'
3484  015267     045  044443  041516  CM2:    .ASCII  '%@INCORRECT PROGRAM SELECTED.@'
3485  015274  051117  042522  052103
3486  015302  050040  047522  051107
3487  015310  046501  051440  046105
3488  015316  041505  042524  027104
3489  015324     100
3490  015325     045  044443  041516  CM3:    .ASCII  '%@INCORRECT ROUTINE SELECTED.@'
3491  015332  051117  042522  052103
3492  015340  051040  052517  044524
3493  015346  042516  051440  046105
3494  015354  041505  042524  027104
3495  015362     100
3496  015363     045  051443  046105  CM4:    .ASCII  '%@SELECT DESIRED SR OPTIONS.'
3497  015370  041505  020124  042504
3498  015376  044523  042522  020104
3499  015404  051123  047440  052120
3500  015412  047511  051516     056
3501  015417     045  047516  046522          .ASCII  '%NORMAL OPERATION IS WITH @ '
3502  015424  046101  047440  042520
3503  015432  040522  044524  047117
3504  015440  044440  020123  044527
3505  015446  044124  040040     040
3506  015453     123  051127  030075  CM4B:   .ASCII  'SWR=000000 @'
3507  015460  030060  030060  020060
3508  015466     100
3509  015467     045  037443  046440  CM5:    .ASCII  '%?? MANUAL ROUTINE, BIT0 (SWREG) IS SET.@'
3510  015474  047101  040525  020114
3511  015502  047522  052125  047111
3512  015510  027105  041040  052111
3513  015516  020070  051450  051127
3514  015524  043505  020051  051511
3515  015532  051440  052105  040056
3516  015540  021445  051120  030107  IM0:    .ASCII  '%@PRG0, READER LOGIC TESTS.@'
3517  015546  020056  042522  042101
3518  015554  051105  046040  043517
3519  015562  041511  052040  051505
3520  015570  051524  040056
3521  015574  021445  051120  031107  IM0A:   .ASCII  '%@PRG2, PUNCH LOGIC TESTS.@'
3522  015602  020056  052520  041516
3523  015610  020110  047514  044507
3524  015616  020103  042524  052123
3525  015624  027123     100
3526  015627     045  050043  043522  IM0B:   .ASCII  '%@PRG3, PUNCH TEST.@'
3527  015634  027063  050040  047125
3528  015642  044103  052040  051505
3529  015650  027124     100
3530  015653     045  050043  043522  IM0C:   .ASCII  '%@PRG11, COUNT PATTERN TAPE GENERA^OR.@'
3531  015660  030461  020056  047503
3532  015666  047125  020124  040520
3533  015674  052124  051105  020116
3534  015702  040524  042520  043440
3535  015710  047105  051105  052101
```

```
3536  016716  051117  040056
3537  016722  051445  052105  052440  IM1:    .ASCII  '%SET UP READER AS FOLLOWS: 0'
3538  016730  020120  042522  042101
3539  016736  051105  040440  020123
3540  016744  047506  046114  053517
3541  016752  035123  040040
3542  016756  047520  042527  020122  IM2:    .ASCII  'POWER OFF, OFF-LINE, NO TAPE.0'
3543  016764  043117  026106  047440
3544  016772  043106  046055  047111
3545  016000  026106  047040  020117
3546  016006  040524  042520  040056
3547  016014  047520  042527  020122  IM3:    .ASCII  'POWER ON, OFF-LINE, NO TAPE.0'
3548  016022  047117  020064  043117
3549  016030  026506  044514  042516
3550  016036  020064  047516  052040
3551  016044  050101  027105     100
3552  016051     120  053517  051105  IM4:    .ASCII  'POWER ON, ON-LINE, NO TAPE.0'
3553  016056  047440  026116  047440
3554  016064  026516  044514  042516
3555  016072  020064  047516  052040
3556  016100  050101  027105     100
3557  016105     045  046120  041501  IM4S:   .ASCII  '%PLACE PRG3 OUTPUT TAPE IN READER, FIRST RUBOUT '
3558  016112  020105  051120  031507
3559  016120  047440  052125  052520
3560  016126  020124  040524  042520
3561  016134  044440  020116  042522
3562  016142  042101  051105  020056
3563  016150  044506  051522  020124
3564  016156  052522  047502  052125
3565  016164     040
3566  016165     123  047510  046125          .ASCII  'SHOULD BE ABOUT 3 INCHES'
3567  016172  020104  042502  040440
3568  016200  047502  052125  031440
3569  016206  044440  041516  042510
3570  016214     123
3571  016215     045  051106  046517          .ASCII  '%FROM RIGHT EDGE OF READER PRESSURE PLATE.%0'
3572  016222  051040  043511  052110
3573  016230  042440  043504  020105
3574  016236  043117  051040  040505
3575  016244  042504  020122  051120
3576  016252  051505  052523  042522
3577  016260  050040  040514  042524
3578  016266  022456     100
3579  016271     120  053517  051105  IM5:    .ASCII  'POWER ON, ON-LINE, TAPE IN READER,0'
3580  016276  047440  026116  047440
3581  016304  026516  044514  042516
3582  016312  020064  040524  042520
3583  016320  044440  020116  042522
3584  016326  042101  051105  040056
3585  016334  046445  045501  020105  IM6:    .ASCII  '%MAKE READER READY,0'
3586  016342  042522  042101  051105
3587  016350  051040  040505  054504
3588  016356  040056
3589  016360  021445  051120  030507  IM7:    .ASCII  '%:PRG1, READER TEST,0'
3590  016366  020056  042522  042101
3591  016374  051105  052040  051505
```

H6

```
3592  016402  027124     100
3593  016405     045  052524  047122  IM10:   .ASCII  'STURN READER OFF-LINE,S'
3594  016412  051040  040505  042504
3595  016420  020122  043117  026506
3596  016426  044514  042516  040056
3597  016434  051445  052105  052440  IM11:   .ASCII  'SET UP PUNCH AS FOLLOWS: S'
3598  016442  020120  052520  041516
3599  016450  020110  051501  043040
3600  016456  046117  047514  051527
3601  016464  020072     100
3602  016467     120  053517  051105  IM12:   .ASCII  'POWER OFF, NO TAPE,S'
3603  016474  047440  043106  020054
3604  016502  047516  052040  050101
3605  016510  027105     100
3606  016513     120  053517  051105  IM13:   .ASCII  'POWER ON, NO TAPE,S'
3607  016520  047440  026116  047040
3608  016526  020117  040524  042520
3609  016534  040056
3610  016536  047520  042527  020122  IM14:   .ASCII  'POWER ON, TAPE IN PUNCH,S'
3611  016544  047117  020054  040524
3612  016552  042520  044440  020116
3613  016560  052520  041516  027110
3614  016566     100
3615  016567     045  042522  047515  IM15:   .ASCII  'SREMOVE TAPE FROM PUNCH,S'
3616  016574  042526  052040  050101
3617  016602  020105  051106  046517
3618  016610  050040  047125  044103
3619  016616  040056
3620  016620  046445  045501  020105  IM16:   .ASCII  'SMAKE PUNCH READY,S'
3621  016626  052520  041516  020110
3622  016634  042522  042101  027131
3623  016642     100
3624  016643     045  050043  043522  IM17:   .ASCII  'S$PRG1$ - READ X, STALL Y,S$S'
3625  016650  030061  026440  051040
3626  016656  040505  020104  026130
3627  016664  051440  040524  046114
3628  016672  054440  022456  040043
3629  016700  021445  051120  032107  IM20:   .ASCII  'S$PRG4, PUNCH VERIFY TEST,'
3630  016706  020056  052520  041516
3631  016714  020110  042526  044522
3632  016722  054506  052040  051505
3633  016730  027124
3634  016732  046045  040517  020104          .ASCII  'SLOAD READER WITH TAPE PRODUCED '
3635  016740  042522  042101  051105
3636  016746  053440  052111  020110
3637  016754  040524  042520  050040
3638  016762  047522  052504  042503
3639  016770  020104
3640  016772  054502  050040  043522          .ASCII  'BY PRG3,S'
3641  017000  027063     100
3642  017003     045  050043  043522  IM21:   .ASCII  'S$PRG6$$S'
3643  017010  022466  040043
3644  017014  021445  051120  033507  IM22:   .ASCII  'S$PRG7$$S'
3645  017022  021445     100
3646  017025     045  051120  051505  IM23:   .ASCII  'SPRESS CONTINUE,S'
3647  017032  020123  047503  052116
```

```
3648  017040  047111  042525  040056
3649  017046  021445  051120  030507  IM24:   .ASCII  '%%PRG12, PTR SPEED TEST,'
3650  017054  027062  050040  051124
3651  017062  051440  042520  042105
3652  017070  052040  051505  027124
3653  017076  046045  040517  020104          .ASCII  '%LOAD ANY TAPE LOOP IN READER '
3654  017104  047101  020131  040524
3655  017112  042520  046040  047517
3656  017120  020120  047111  051040
3657  017126  040505  042504  020122
3658  017134  047101  020104  040515          .ASCII  'AND MAKE READY,%%0'
3659  017142  042513  051040  040505
3660  017150  054504  022466  040043
3661  017156  050045  042522  051523  IM24A:  .ASCII  '%PRESS CONTINUE TO START TIMING,%%0'
3662  017164  041440  047117  044524
3663  017172  052516  020105  047524
3664  017200  051440  040524  052122
3665  017206  052040  046511  047111
3666  017214  027107  021445     100
3667  017221     045  050043  043522  IM25:   .ASCII  '%%PRG13, PTP SPEED TEST,0'
3668  017226  031461  020056  052120
3669  017234  020120  050123  042505
3670  017242  020104  042524  052123
3671  017250  040056
3672  017252  021445  051120  032507  IM26:   .ASCII  '%%PRG5, COMBINED READER-PUNCH TEST,'
3673  017260  020056  047503  041115
3674  017266  047111  042105  051040
3675  017274  040505  042504  026522
3676  017302  052520  041516  020110
3677  017310  042524  052123     056
3678  017315     045  040515  042513          .ASCII  '%MAKE PUNCH READY, PUNCH BLANK LEADER, '
3679  017322  050040  047125  044103
3680  017330  051040  040505  054504
3681  017336  020054  052520  041516
3682  017344  020110  046102  047101
3683  017352  020113  042514  042101
3684  017360  051105  020054
3685  017364  047514  042101  044440          .ASCII  'LOAD IN READER,0'
3686  017372  020116  042522  042101
3687  017400  051105  040056
3688  017404  051045  040505  042504  SM1:    .ASCII  '%READER ERROR BIT SET,0'
3689  017412  020122  051105  047522
3690  017420  020122  044502  020124
3691  017426  042523  027124     100
3692  017433     045  042522  042101  SM2:    .ASCII  '%READER NOT READY,0'
3693  017440  051105  047040  052117
3694  017446  051040  040505  054504
3695  017454  040056
3696  017456  021445  052520  041516  SM3:    .ASCII  '%%PUNCH NOT READY,0'
3697  017464  020110  047516  020124
3698  017472  042522  042101  027131
3699  017500     100
3700  017501     045  051043  040505  SM4:    .ASCII  '%%READER SPEED : 0'
3701  017506  042504  020122  050123
3702  017514  042505  020104  020072
3703  017522     100
```

DIPCAE,SRC

```
3704  017523     045  050043  047125  SM5:    .ASCII  '%PUNCH SPEED : 0'
3705  017530  046103  051440  042520
3706  017536  042105  035040  040040
3707  017544  020040  020040  041440  ACPS:   .ASCII  '     CHARS PER SEC.,0'
3708  017552  040510  051522  050040
3709  017560  051105  051440  041505
3710  017566  040056
3711  017570  021445  051105  047522  EM6:    .ASCII  '%ERROR P '
3712  017576  020122  020120
3713  017602  020040  020040  020040  APNUMB: .ASCII  '      T '
3714  017610  020124
3715  017612  020040  020040  020040  ATNUMB: .ASCII  '      PC '
3716  017620  041520     040
3717  017623     040  020040  020040  APC:    .ASCII  '      0'
3718  017630  040040
3719  017632  020040  040504  040524  EM1:    .ASCII  '  DATA ERROR  S/B: '
3720  017640  042440  051122  051117
3721  017646  020040  027523  035102
3722  017654     040
3723  017655     040  020040  020040  ASB:    .ASCII  '      WAS: '
3724  017662  053440  051501  020072
3725  017670  020040  020040     100  AWAS:   .ASCII  '    0'
3726  017675     040  042522  042522  EM2:    .ASCII  ' REREAD ERROR,  1ST READ: '
3727  017702  042101  042440  051122
3728  017710  051117  020056  030440
3729  017716  052123  051040  040505
3730  017724  035104     040
3731  017727     040  020040  020040  ORGRD:  .ASCII  '      WAS: '
3732  017734  053440  051501  020072
3733  017742  020040  020040     100  SUBRD:  .ASCII  '    0'
3734  017747     040  054523  041516  EM3:    .ASCII  ' SYNC ERROR,0'
3735  017754  042440  051122  051117
3736  017762  040056
3737  017764  046045  040505  042504  EM4:    .ASCII  '%LEADER ERROR,  S/B: '
3738  017772  020122  051105  047522
3739  020000  027122  020040  027523
3740  020006  035102     040
3741  020011     040  020040  020040  ESB:    .ASCII  '      WAS: '
3742  020016  053440  051501  020072
3743  020024  020040  020040     100  EWAS:   .ASCII  '    0'
3744  020031     045  042514  042101  EM5:    .ASCII  '%LEADER ERROR, S/B BETWEEN '
3745  020036  051105  042440  051122
3746  020044  051117  020056  027523
3747  020052  020102  042502  053524
3748  020060  042505  020116
3749  020064  020060  047101  020104          .ASCII  '0 AND 3, WAS : '
3750  020072  027063  053440  051501
3751  020100  035040     040
3752  020103     040  020040  040040  FWAS:   .ASCII  '    0'
3753  020110  046440  052101  044103  EM6:    .ASCII  ' MATCH ERR,0'
3754  020116  042440  051122  040056
3755                                          .EVEN
3756  020124  003407                  EM7:    3407                    ;DOUBLE BELL,
3757  020126  021445  052120  020122          .ASCII  '%PTR NRP0'
3758  020134  051116  040120
3759  020140  043040  046101  042523  EM10:   .ASCII  ' FALSE RDR, INTR0'
```

```
3760  020146  051040  051104  020056
3761  020154  047111  051124     100
3762  020161     040  040506  051514  EM11:   .ASCII  ' FALSE PUN INTR@'
3763  020166  020105  052520  020116
3764  020174  047111  051124     100
3765  020201     045  051120  031507  PJEND:  .ASCII  '%PRG) END OF PASS@'
3766  020206  042440  042116  047440
3767  020214  020106  040520  051523
3768  020222     100
3769  020223     045  043536  040045  $CTLG:  .ASCII  '%^G%@'
3770  020230  040045                  $CRLF:  .ASCII  '%@'
3771  020232  037445  040043          $QUEST: .ASCII  '%?#@'
3772  020236  020040  020040  020040  $VALUE: .ASCII  '       @'
3773  020244  040040
3774  020246  051445  046105  041505  $RTN:   .ASCII  '%SELECT ROUTINE NO, @'
3775  020254  020124  047522  052125
3776  020262  047111  020105  047516
3777  020270  020056     100
3778  020273     040  047040  053505  $NEW:   .ASCII  '  NEW= @'
3779  020300  020075  040040
3780  020304  021445  053523  036522  $SWREQ: .ASCII  '%$SWR= @'
3781  020312  040040
3782  020314  042445  052116  051105  $STEST: .ASCII  '%ENTER PROGRAM NO, @'
3783  020322  050040  047522  051107
3784  020330  046501  047040  027117
3785  020336  040040
3786  020340  042445  052116  051105  $TIME:  .ASCII  '%ENTER TIMING @'
3787  020346  052040  046511  047111
3788  020354  020107     100
3789  020357     045  047105  042524  $STALL: .ASCII  '%ENTER STALL @'
3790  020364  020122  052123  046101
3791  020372  020114     100
3792  020375     045  047105  042524  $NUMCR: .ASCII  '%ENTER CHARACTER COUNT @'
3793  020402  020122  044103  051101
3794  020410  041501  042524  020122
3795  020416  047503  047125  020124
3796  020424     100
3797  020425     045  051461  020124  $CH1:   .ASCII  '%1ST CHAR TO PUNCH (ASCII) = @'
3798  020432  044103  051101  052040
3799  020440  020117  052520  041516
3800  020446  020110  040450  041523
3801  020454  044511  020051  020075
3802  020462     100
3803  020463     045  047062  020104  $CH2:   .ASCII  '%2ND CHAR TO PUNCH (ASCII) = @'
3804  020470  044103  051101  052040
3805  020476  020117  052520  041516
3806  020504  020110  040450  041523
3807  020512  044511  020051  020075
3808  020520     100
3809  020521     045  050040  042522  $$KEY:  .ASCII  '% PRESS CONTINUE WHEN READY%@'
3810  020526  051523  041440  047117
3811  020534  044524  052516  020105
3812  020542  044127  047105  051040
3813  020550  040505  054504  021445
3814  020556  052123  044522  042513          .ASCII  'STRIKE ANY KEY AT END OF TIMING%@'
3815  020564  040440  054516  045440
```

DZPCAE,SRC

```
3016  020572  054505  040440  020124
3017  020600  047105  020104  043117
3018  020606  052040  046511  047111
3019  020614  022507     100
3020  020617     045  051461  020124  $RD1:   .ASCII  '%1ST CHAR TO READ (ASCII) = 0'
3021  020624  044103  051101  052040
3022  020632  020117  042522  042101
3023  020640  024040  051501  044503
3024  020646  024511  036440  040040
3025  020654  031045  042116  041440  $RD2:   .ASCII  '%2ND CHAR TO READ (ASCII) = 0'
3026  020662  040510  020122  047524
3027  020670  051040  040505  020104
3028  020676  040450  041523  044511
3029  020704  020051  020075     100
3030  020711     045  050043  036503  PCHLT:  .ASCII  '%0PC= '
3031  020716     040
3032  020717     040  020040  020040  GWAS:   .ASCII  '       -HALT%00'
3033  020724  020040  044055  046101
3034  020732  022524  040043
3035  020736  025045  025052  047105  ENDRTN: .ASCII  '%***END-RTN NO, '
3036  020744  026504  052122  020116
3037  020752  047516  020056
3038  020756  020040  020040  026440  RTNN:   .ASCII  '     -HALT%00'
3039  020764  040510  052114  021445
3040  020772     100
3041  020773     045  046443  044501  $TITLE: .ASCII  '%$MAINDEC-11-DZPCA-E%'
3042  021000  042116  041505  030455
3043  021006  026461  055104  041520
3044  021014  026501  022505
3045  021020  041520  030461  051040          .ASCII  'PC11 READER-PUNCH TEST8%00'
3046  021026  040505  042504  026522
3047  021034  052520  041516  020110
3048  021042  042524  052123  022523
3049  021050  040043
3050          000001                          .END
```

DZPCAE,SRC SYMBOL TABLE

```
ACNV    005112
ACNVB   005046
ACNVC   005074
ACNVM   005126
ACNVX   005110
ACNV4   005054
ACNV6   005026
ACPS    017544
ADTEMP  005364
APC     017623
APGEND  015263
APNUMB  017602
ARDA    002672
ARDB    002710
ARDER   002530
AREAD   002662
AREAD1  002666
ARRDY   002642
ARRDYA  002654
ASB     017655
ATNUMB  017612
AT0     005432
AT0A    005450
AT0E    005456
AT1     005462
AT1A    005500
AT1E    005506
AT10    006104
AT10A   006114
AT10E   006134
AT11    006140
AT11A   006162
AT12    006174
AT12A   006204
AT12E1  006236
AT12E2  006242
AT13    006246
AT13A   006256
AT14    006314
AT14A   006324
AT14C   006366
AT15    006376
AT15A   006406
AT15E   006442
AT16    006446
AT16A   006456
AT16B   006506
AT16E   006524
AT17    006530
AT17A   006540
AT17B   006556
AT17E   006602
AT2     005512
AT2A    005546
AT20    006630
AT20A   006650
AT20B   006670
AT20X   006740
AT21    006744
AT21A   006760
AT21B   007012
AT21E   007010
AT22    007014
AT22A   007030
AT22E   007064
AT23    007070
AT23A   007104
AT23B   007144
AT23E   007142
AT24    007146
AT24A   007156
AT24C   007214
AT24D   007230
AT24E1  007210
AT24E2  007234
AT25    007240
AT25A   007254
AT25B   007314
AT25E   007312
AT26    007316
AT26A   007350
AT26B   007410
AT26E1  007402
AT26E2  007406
AT27    007412
AT27A   007444
AT27C   007500
AT27D   007514
AT27E1  007524
AT27E2  007474
AT27E3  007520
AT3     005562
AT3A    005616
AT30    007530
AT30A   007576
AT30B   007624
AT30C   007632
AT30D   007654
AT30E1  007656
AT30E2  007620
AT30E3  007652
AT4     005630
AT4A    005672
AT5     005704
AT5A    005746
AT6     005760
AT6A    005776
AT6B    006020
AT6E1   006014
AT7     006042
AT7A    006060
AWAS    017670
A1ST    005102
BCHECK  004372
BDCNV   005256
BDCNVA  005276
BELL    = 000007
BIT0    = 000000
BIT1    = 000002
BIT10   = 002000
BIT11   = 004000
BIT12   = 010000
BIT13   = 020000
BIT14   = 040000
BIT15   = 100000
BIT2    = 000004
BIT3    = 000010
BIT4    = 000020
BIT5    = 000040
BIT6    = 000100
BIT7    = 000200
BIT8    = 000400
BIT9    = 001000
BNOVA   005174
BNOVE   005166
BRCTR   001242
BRDBB   003210
BRDCC   003222
BRDDD   003230
BREAD   003102
BREADA  003144
BREADB  003162
BREADC  003216
BSYNC   004452
BT0     007724
BT0A    007740
BT1     007752
BT1A    007774
BT2     010010
BT2A    010032
BT2C    010042
BT3     010062
BT3A    010112
BT3C    010120
BT4     010140
BT4A    010170
BT4C    010176
CC      = 177776
CHAIN   002110
CHALT   = 104410
CHLT    002462
CHNAA   002204
CHNB    002244
CHR1    001402
CHR1A   001410
CHR2    001404
CHR2A   001412
CHR3    001406
CHR3A   001414
CKSWR   = 104020
CKSWRR  015120
CLEAN   001716
CM2     015267
CM3     015325
CM4     015363
CM4B    015453
CM5     015467
CNTL    = 104015
CNTLU   015156
CNVCTR  005356
COUNT   001372
CPKPL   014616
CPKPLA  014644
CPRDY   005204
CPRDYA  005222
CRBUF   001400
CRDA    013122
CRDAA   013134
CREAD   013106
CREADA  013174
CREADB  013216
CREADC  013226
CRIA    001672
CRIB    001702
CTRA    001420
CTRB    001422
CTRC    001424
CTRD    001426
CT0     010240
CT0A    010256
CT0E    010264
CT1     010270
CT1A    010306
CT1E    010314
CT10    010652
CT10A   010662
CT10B   010670
CT10C   010734
CT11    010742
CT11A   010752
CT11E   011000
CT12    011004
CT12A   011020
CT12C   011050
CT12E   011046
CT13    011052
CT13A   011062
CT13C   011120
CT13D   011134
CT13E1  011114
CT13E2  011140
CT14    011144
CT14A   011160
CT14E   011214
CT15    011220
CT15A   011234
CT15B   011274
CT15E   011272
CT16    011276
CT16A   011312
CT16B   011354
CT16E   011352
CT17    011356
CT17A   011410
CT17B   011454
CT17C   011470
CT17E1  011450
CT17E2  011474
CT17E3  011500
CT2     010320
CT2A    010354
CT20    011504
CT20A   011536
CT20B   011572
CT20C   011600
CT20D   011622
CT20E1  011624
CT20E2  011566
CT20E3  011620
CT3     010370
CT3A    010424
CT4     010436
CT4A    010472
CT5     010504
CT5A    010514
CT5B    010544
CT6     010566
CT6A    010576
CT7     010630
CT7A    010640
CUPTST  001254
DECVAL  015256
DELAY   = 104000
DELAYX  = 104400
DIGIT   005360
```

A7

DIPCAE,SRC SYMBOL TABLE

```
DISPLA  001206
DISPRE  000174
DLCNT   003716
DLCTR   003714
DLY     003566
DLYA    003610
DLYB    003616
DLYX    004310
DLYXA   004322
DLYXB   004330
DLYX0 = 004314
DLYX1 = 004326
DT0     011662
DT0A    011672
DT0B    011712
DT1     011732
DT1A    011750
DT1B    011770
DT2     012012
DT2A    012036
DT2B    012056
DT2C    012064
DT2D    012112
DT3     012114
DT3A    012124
DT3B    012144
DT3C    012156
DT3D    012204
DT4     012206
DT4A    012216
DVDND   001244
DVQUOT  001246
ECHK    012622
ECHKA   012656
EHALT = 104001
EHLT    002514
EHLTA   002526
EMTINT  002442
EMTTAB  001320
EMTX  = 000021
EM0     017570
EM1     017632
EM10    020140
EM11    020161
EM2     017675
EM3     017747
EM4     017764
EM5     020031
EM6     020110
EM7     020124
ENDRTN  020736
ERCTR   001416
ERR     003234
ERRA    003320
ERROR = 104006
ERROR1= 104007
ERRT    001362
ERR1    003244
ERR1A   003272
ESB     020011
ET0A    012330
ET0B    012336
ET0C    012344
ET0D    012374
ET0E    012402
ET0F    012410
ET0G    012432
ET0H    012476
ET0I    012530
ET0J    012542
ET0K    012572
EWAS    020024
FORWD   002364
FRST    001370
FWAS    020103
GETRDY  001710
GOTST   002416
GOTSTA  002436
GRCNT   004346
GTBIN   004712
GTBINP  004760
GTRDYA  001736
GTRDYB  001742
GTRDYC  001760
GTRDYD  002062
GWAS    020717
HERE    002350
HSPCN   005236
HT0A    013720
HT0B    013770
HT0C    013776
HT0D    014030
HT0E    014054
HT0F    014104
HT0G    014126
ICTR    001262
IM0     015540
IM0A    015574
IM0B    015627
IM0C    015653
IM1     015722
IM10    016405
IM11    016434
IM12    016467
IM13    016513
IM14    016536
IM15    016567
IM16    016620
IM17    016643
IM2     015756
IM20    016700
IM21    017003
IM22    017014
IM23    017025
IM24    017046
IM24A   017156
IM25    017221
IM26    017252
IM3     016014
IM4     016051
IM4S    016105
IM5     016271
IM6     016334
IM7     016360
INBIN   004654
INCRTN  002100
INGXOR  001570
INHPRT  003300
ITA     014246
ITB     014272
ITX     014320
ITY     014314
KSTART  001252
KTA     014416
KTB     014450
KTC     014460
KTD     014476
KTE     014504
KTF     014512
LOGIC   002340
LTA     014540
LTB     014554
LTC     014572
LTD     014600
LTE     014606
MACHER  000004
MANUAL= 100000
MESS    002042
MSEC    001250
NOP   = 000240
NTYET   002016
NXTST   001260
OPEN  = 000000
OPTS    014722
OPTSEL= 104014
ORGRD   017727
OUT     015240
PBIN    012766
PBINA   013070
PBINB   013104
PBNA    013004
PBNB    013016
PC    =%000007
PCMCNT  013234
PCNLT   020711
PCNLVL  001226
PCNVTR  001224
PCSIN   002554
PFRNT   012226
PIND    004704
POPSP = 005726
POPSP2= 022626
PPB     001216
PPS     001214
PRB     001212
PRGID   001266
PRGNUM  001240
PRGTAB  001270
PRG0    005376
PRG1    007666
PRG10   014134
PRG11   014322
PRG11A  014344
PRG11B  014360
PRG12   014372
PRG13   014524
PRG2    010216
PRG3    011634
PRG4    012302
PRG5    012660
PRG6    013454
PRG6A   013562
PRG7    013606
PRS     001210
PRTY0 = 000000
PRTY1 = 000040
PRTY2 = 000100
PRTY3 = 000140
PRTY4 = 000200
PRTY5 = 000240
PRTY6 = 000300
PRTY7 = 000340
PSW   = 177776
PTINTA  004254
PTMCAL  004170
PTMERR  004076
PTMINT  004234
PT0     004700
PT0P    004706
PT1     004702
PT1P    004710
PUNC1   013604
P3END   020201
RBIN    013272
RBINA   013276
RBINB   013312
RBINC   013352
RBIND   013402
RBUSY   013236
RCMSK   004366
RCNT    001376
RDRLVL  001222
RDRVTR  001220
RETRN   002634
RIND    004676
RMCNT   004370
RNGEN   003030
RP1     003076
RP2     003100
RTINTA  004116
RTINTB  004134
RTINTC  004152
RTMCAL  003720
RTMCLA  003750
RTMCLB  003754
RTMERR  004064
RTMINT  004044
RTRN    020756
RTRNO   001256
R0    =%000000
R1    =%000001
R2    =%000002
R3    =%000003
R4    =%000004
R5    =%000005
R6    =%000006
SCOPE = 104013
SCOPTR  001264
SM1     017404
SM2     017433
SM3     017456
SM4     017501
SM5     017523
SP    =%000006
SPBUT = 001200
SRESET= 104002
SRSETT  003012
SRTN    020246
STAL    004262
STALA   004302
STALB   004304
STALL = 104005
START   001432
STLMSK  004306
STPCHV= 104012
```

DZPCAE,SRC SYMBOL TABLE

```
$TPPA   003000          $YNCB   004550          T$M2    002716          TYPSB   003562          $CALF   020230
$TPRA   002750          $YNCC   004636          TTIN    014654          VALID = 104017          $CTLG   020223
$TPTPV  002762          $18     002156          TTYIN = 104016          VALINP  014740          $NEW    020273
$TPTRV  002732          $28     002166          TYP     003400          WHZERO  013240          $NUMCR  020375
$TRDRV= 104011          TENPWR  005362          TYPA    003410          XCNT    001430          $QUEST  020232
SUBRD   017742          TIB     001374          TYPC    003440          XCT     006362          $RD1    020617
SUBTEN  005316          TKB     001232          TYPD    003466          XOR     002354          $RD2    020654
SUBTNA  005322          TKS     001230          TYPDAT  003532          XORA    001662          $SKEY   020521
SUBTNB  005336          TLX     014724          TYPE  = 104003          XORFLG  002040          $STALL  020357
SWR     001204          TMCON = 004136          TYPES = 104004          XP      010730          $STEST  020314
SWREG   000176          TMP1    001364          TYPF    003504          XPBE    010736          $SWREG  020304
SWTL    015242          TMP2    001366          TYPG    003516          XTP     006742          $TIME   020340
S.CTRA  004652          TPB     001236          TYPS    003534          $CH1    020425          $TITLE  020773
SYNCA   004542          TPS     001234          TYPSA   003560          $CH2    020463          $VALUE  020236
.     = 021052
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

@,DZPCAE/SOL=DZPCAE,SRC
RUN-TIME:  11 23 1 SECONDS
RUN-TIME RATIO:  75/37=2.0
CORE USED:  6K  (11 PAGES)
```