VT03

DISPLAY TEST

MD-11-DZVTB-D

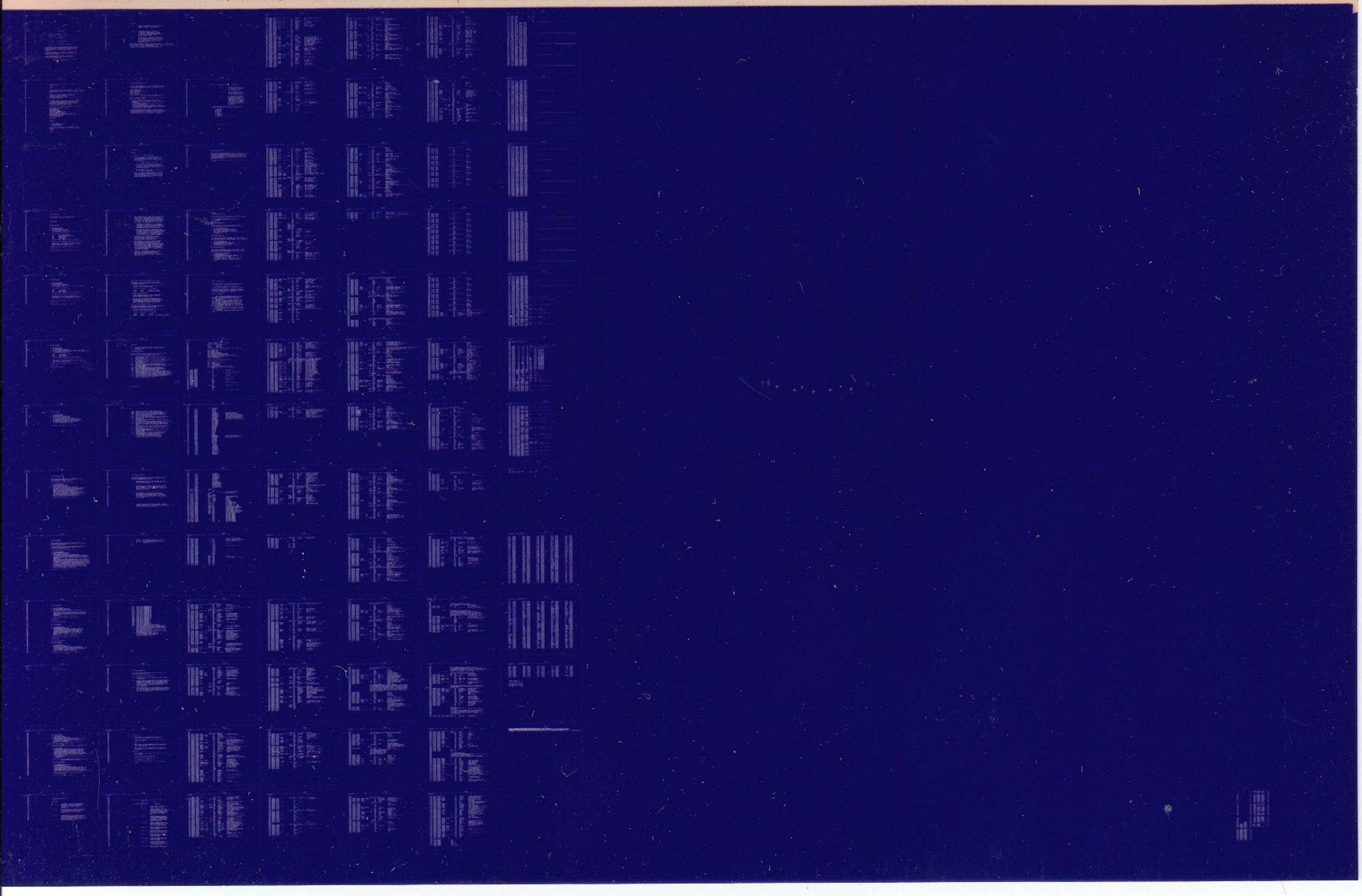
EP-DZVTB-D-DL-B

FICHE 1 OF 1

DEC 1976

digital

MADE IN USA



```
                                .REPT 0
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZVTB-D-D

PRODUCT NAME: VTDS DISPLAY TERMINAL TESTS

DATE: DECENMBER,1976

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JIM CARRON/VIN KALINOWSKI

NOTE: THIS PROGRAM IS A MODIFIED VERSION OF THE TELETYPE DIAGNOSTIC
MAINDEC-11-D2AA

1. ABSTRACT

THE VT05 DISPLAY TERMINAL TESTS CONSISTS OF A PACKAGE OF TEST PROGRAMS DESIGNED TO TEST THE VT05 INPUT-OUTPUT LOGIC, THE VT05 DISPLAY, AND THE KEYBOARD. ALL TESTS ARE INCLUDED IN ONE OBJECT TAPE.

THIS TEST CAN TEST THE VT05 WHEN INTERFACED BY EITHER A SINGLE (KL) OR DOUBLE BUFFERED (DL) CONTROLLER COMMUNICATING SERIAL TO 2400 BAUD.
UNDER MONITOR LOAD IT WILL TEST THE CONSOLE DEVICE AND ALL CONTIGUOUS EXTRA DEVICES.

THE PROGRAM USES THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES IF NO HARDWARE SWITCH REGISTER IS FOUND.
THE OPERATOR IS RESPONSIBLE FOR LOADING THE DESIRED VALUE BEFORE STARTING THE PROGRAM. LOCATION 174 WILL BE USED AS THE SOFTWARE DISPLAY REGISTER.

THE AVAILABLE TEST PROGRAMS ARE LISTED HERE IN NUMERICAL ORDER:

PRG0-COMBINED INPUT-OUTPUT LOGIC TESTS
PRG1-DISPLAY TEST
PRG2-KEYBOARD TEST
PRG3-PRINTER EXERCISER
PRG4-CLOCK ADJUSTMENT ROUTINE
PRG5-CLOCK ADJUSTMENT ROUTINE
PRG6-MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
PRG7-MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN TEST.
PRG10-ROLL-UP DISPLAY TEST
PRG11-CURSOR ADDRESS TEST

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11. SYSTEM. (4 K CORE).
B. VT05 DISPLAY TERMINAL
C. HIGH SPEED READER

THE VT05 MUST HAVE STANDARD TELETYPE ADDRESSES. REFER TO SECTION 7.3 IF THE VT05 DOES NOT HAVE STANDARD PERIPHERAL ADDRESSES.

2.2 STORAGE

```
                                        THIS PROGRAM USES LOCATION 00200 THROUGH 17434.
```

3. LOADING PROCEDURE

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

4. USE PROCEDURE

4.1 PRGO USE PROCEDURE

A. SET VT05 TO ON-LINE.
B. LOAD ADDRESS 000200
C. SET SR TO 000000. PRESS START
D. THE PROGRAM STOPS AT COMMON HALT.
E. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.

THIS PROGRAM'S SR OPTIONS ARE:

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

SECTION 7.2 GIVES A COMPLETE EXPLANATION OF SR OPTIONS.

F. PRESS CONTINUE. THE PROGRAM IS EXECUTED AND STOPS AT PROGRAM END HALT WHEN COMPLETED, PROVIDED NO ERRORS OCCUR.
G. REFER TO SECTION 6. IF ERRORS OCCUR.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.1

EXECUTION TIME:

ONE NORMAL ERROR FREE PASS TAKES APPROXIMATELY 4 MINUTES.

```
151
152
153                                 4.2     PRG1 USE PROCEDURE
154                                         ------------------
155
156                                         A.  SET VT05 TO ON-LINE
157                                         B.  LOAD ADDRESS 000200.
158                                         C.  SET SR TO 000001.  PRESS START
159                                         D.  PROGRAM STOPS AT COMMON HALT.
160                                         E.  SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.
161
162                                             THIS PROGRAM'S SR OPTIONS ARE:
163
164                                             SR15          HALT AT END OF ROUTINE
165                                             SR10          LOOP PROGRAM
166                                             SR9           SELECT ROUTINE
167                                             SR6 THROUGH SR0 = NUMBER OF ROUTINE TO BE SELECTED.
168
169                                             SECTION 7.2 GIVES A COMPLETE EXPLANATION OF SR OPTIONS.
170
171                                         F.  PRESS CONTINUE.  THE VT05 WILL BE EXERCISED AND
172                                             THE PROGRAM WILL STOP AT PROGRAM END HALT WHEN COMPLETED.
173                                         G.  ERROR DETECTION IS BY VISUAL INSPECTION OF DISPLAY.
174
175
176                                         A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION  8.2
177
178                                         EXECUTION TIME:
179
180                                         ONE NORMAL PASS TAKES APPROXIMATELY 12 MINUTES.
```

4.3 PRG2 USE PROCEDURE

A. SET VT05 ON-LINE.
B. LOAD ADDRESS 000200.
C. SET SR TO 000002. PRESS START
D. THE PROGRAM TYPES "KEYBOARD TEST" AND STOPS AT COMMON HALT.
E. SET ANY DESIRED SR OPTIONS. NORMAL RUN IS WITH SR = 000000.
   THIS PROGRAM'S SR OPTIONS ARE:

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

   SECTION 7.2 GIVES A COMPLETE EXPLANATION OF SR OPTIONS.

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

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.3

EXECUTION TIME:

PROGRAM IS USER DEPENDENT.

```
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
```

4.4 PRG3 USE PROCEDURE

A. SET VT05 TO ON-LINE
B. LOAD ADDRESS 000200
C. SET SR TO 000003. PRESS START
D. THE PROGRAM TYPES "TYPE IN DATA"
E. KEY IN ANY FIVE CHARACTERS TO BE TYPED.
F. KEY IN EITHER A RUBOUT FOR FULL SPEED TYPING, OR ANY OTHER CHARACTER FOR RANDOM STALLS BETWEEN CHARACTERS.
G. THE PROGRAM TYPES CONTINUOUSLY LINES CONTAINING THE FIVE CHARACTERS SPECIFIED, UNTIL SR15 IS SET TO A 1. AT THAT POINT THE PROGRAM GOES TO STEP E.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.8

4.5 PRG4 USE PROCEDURE

PRG4 IS USED AS AN AID IN ADJUSTING THE TRANSMITTER CLOCK, AND IN OBSERVING THE DATA BITS AS THEY ARE SHIFTED OUT OF THE TRANSMITTER BUFFER. A SCOPE IS REQUIRED.

TO ADJUST THE PUNCH CLOCK PROCEED AS FOLLOWS:

A. LOAD ADDRESS 000200
B. SET SR TO 00004. PRESS START.
C. PROGRAM STOPS AT COMMON HALT.
D. SET ANY DESIRED ASCII CODE IN LEFT HALF OF SR.
E. SET NUMBER OF MILLISECONDS TO DELAY BETWEEN PUNCH COMMANDS IN RIGHT HALF OF SR. THE NUMBER OF MILLISECONDS SELECTED SHOULD BE LONG ENOUGH FOR THE ENTIRE PUNCH OPERATION TO COMPLETE. A SUGGESTED STARTING NUMBER IS 177.
F. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY. FIRST IT LOADS THE PUNCH BUFFER WITH THE CHARACTER IN SR LEFT, AND THEN DELAYS FOR THE NUMBER OF MILLISECONDS SPECIFIED IN SR RIGHT BEFORE RELOADING THE PUNCH BUFFER AGAIN.
G. SET UP A SCOPE AND DISPLAY THE PUNCH CLOCK PULSES. ADJUST THE PUNCH CLOCK ACCORDING TO SPECIFICATIONS.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.4

4.6 PRG5 USE PROCEDURE

PRG5 IS USED AS AN AID IN ADJUSTING THE RECEIVER CLOCK, AND IN OBSERVING THE DATA BITS AS THEY ARE SHIFTED INTO THE RECEIVER BUFFER. A SCOPE IS REQUIRED.

THE PROGRAM MAKES USE OF THE TRANSMIT MAINTENANCE BIT FEATURE IN ORDER TO CAUSE THE DATA OUTPUTTED TO THE TRANSMITTER BUFFER TO BE SHIFTED INTO THE RECEIVER BUFFER.

TO ADJUST THE RECEIVER CLOCK PROCEED AS FOLLOWS:

A. LOAD ADDRESS 000200
B. SET SR TO 000005. PRESS START.
C. PROGRAM STOPS AT COMMON HALT.
D. SET ANY DESIRED ASCII CODE IN LEFT HALF OF SR.
E. SET NUMBER OF MILLISECONDS TO DELAY BETWEEN TRANSMIT COMMANDS IN RIGHT HALF OF SR. THE SELECTED NUMBER SHOULD BE LONG ENOUGH FOR THE ENTIRE TRANSMIT/RECEIVE OPERATION TO COMPLETE. A SUGGESTED STARTING NUMBER IS 177.
F. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY. FIRST IT LOADS THE TRANSMITTER BUFFER WITH THE CHARACTER IN SR LEFT, AND THEN DELAYS THE NUMBER OF MILLISECONDS SPECIFIED IN SR RIGHT. AS THE DATA BITS ARE SHIFTED OUT OF THE TRANSMITTER BUFFER, THE RECEIVER CLOCK STARTS, AND THE DATA BITS ARE SHIFTED INTO THE RECEIVER BUFFER. AT THE END OF THE DELAY THE PROGRAM MOVES THE RECEIVER BUFFER CONTENTS TO REG 0, AND ISSUES 5 RESET INSTRUCTIONS IN ORDER TO MAKE THE RECEIVER BUFFER CONTENTS VISIBLE IN THE RIGHT HALF OF THE DATA LIGHTS.
G. SET UP A SCOPE AND DISPLAY THE RECEIVER CLOCK PULSES. ADJUST THE RECEIVER CLOCK ACCORDING TO SPECIFICATIONS.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.5

4.7 PRG6 USE PROCEDURE

A. LOAD ADRESS 000200.
B. SET SR TO 000006. PRESS START
C. THE PROGRAM STOPS AT COMMON HALT.
D. SET CODE FOR CHARACTER TO BE TESTED IN THE LEFT HALF OF THE SR.
E. PRESS CONTINUE. THE PROGRAM RUNS CONTINUOUSLY, OUTPUTING THE CHARACTER TO THE OUTPUT BUFFER AND CHECKING THAT THE RECEIVE BUFFER CONTAINS THE SAME CHARACTER WHEN THE RECEIVE DONE BIT BECOMES SET.
F. REFER TO SECTION 6. ERRORS, IF ERRORS OCCUR.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.6

EXECUTION TIME:

CONTINUOUS RUNNING PROGRAM.

4.8 PRG7 USE PROCEDURE

A. LOAD ADDRESS 000200.
B. SET SR TO 00007. PRESS START
C. THE PROGRAM RUNS CONTINUOUSLY. THE SPECIAL BINARY COUNT PATERN IS OUTPUTTED TO THE OUTPUT BUFFER. EACH TIME THE RECEIVE DONE BIT BECOMES SET THE CHARACTER IN THE RECEIVE BUFFER IS CHECKED TO SEE THAT IT MATCHES THE PREVIOUSLY OUTPUTTED CHARACTER. THE PROGRAM STALLS RANDOMLY BETWEEN CHARACTERS. TO RUN AT FULL SPEED, SET SR8 TO A 1.
D. REFER TO SECTION 6. ERRORS, IF ERRORS OCCUR.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.7

EXECUTION TIME:

CONTINUOUS RUNNING PROGRAM.

4.9 PRG10 USER PROCEDURE

A. LOAD ADDRESS 000200.
B. SET SR TO 000010. PRESS START
C. PROGRAM RUNS CONTINUOUSLY. THE SCREEN IS FILLED WITH ALTERNATE LINES OF A CHARACTER AND ITS COMPLEMENT AND A LINE OF THE COMPLEMENT OF THE CHATACTER FOLLOWED BY THE CHARACTER. THIS TEST VERIFIES THE ROLL-UP CAPABILITY OF THE VT05.
D. REFER TO SECTION 6. ERRORS, IF ERRORS OCCUR.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.10

```
348                                     EXECUTION TIME:
349
350                                     CONTINUOUS RUNNING PROGRAM.
```

```
351
352
353                     4.10    PRG11 USER PROCEDURE
354                             --------------------
355                             A.  LOAD ADDRESS 000200.
356                             B.  SET SR TO 000011. PRESS START
357                             C.  CURSOR ADDRESSING CODES WILL BE UTILIZED TO RANDOMLY COVER
358                              THE FACE OF THE CRT WITH THE APPROPRIATE MESSAGE.
359                                 AT THIS POINT THE PROGRAM WILL HALT, A RERUN IS POSSIBLE BY
360                                 DEPRESSING THE CONTINUE KEY.
361                             A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.11
362
363                     EXECUTION TIME: BAUD DEPENDANT, APPROX. 3-5 MIN.
364
365                     4.11    MONITOR LOAD USE PROCEDURE - ACT11 OR DDP1 - NON PAPER TAPE SYSTEM
366                             ------------------------------------------------------------------
367
368                             A.  SET SR TO 000000
369                             B.  WHEN LOADED BY THE MONITOR THIS TEST WILL AUTOMATICALLY
370                                 EXECUTE PROG. 0 AND PROG 1 FOR THE CONSOLE DEVICE AND FOR
371                                 ALL CONTIGUOUSLY ASSIGNED EXTRA DEVICES.  IN ADDITION
372                                 PROGRAM 0 RUNS PROGRAM 7 AS A SUBROUTINE TO VERIFY THE
373                                 CORRECT TRANSFER OF DATA THROUGH THE INTERFACE AND PROGRAM 1
374                                 RUNS PROGRAM 11 AS A SUBROUTINE TO VERIFY THE CURSOR ADDRESSING FEATURES
375                             C.  REFER TO SECTION 6, ERRORS.  ERRORS DETECED ARE LOGGED
376                                 ON THE CONSOLE DEVICE
377                             D.  AT THE END OF A PASS OF ALL DEVICES CONTROL IS RETURNED
378                                 TO THE MONITOR
379
380                             EXECUTION TIME:  BAUD AND NUMBER OF DEVICES DEPENDENT.  APPROXIMATELY
381                                              20 MIN FOR ERROR FREE PASS PER DEVICE
382
383                     4.12    GROUP TEST USE PROCEDURE
384                             ------------------------
385
386                             A.  LOAD CONSTANTS AS DESCRIBED IN 7.4
387                             B.  LOAD ADDRESS 000200
388                             C.  SET SR TO 000000.  PRESS START
389                             D.  THIS TEST WILL AUTOMATICALLY EXECUTE PROG. 0 AND PROG 1
390                                 FOR THE GROUP OF EXTRA DEVICES AS DEFINED IN 7.4 IN ADDITION
391                                 PROGRAM 0 RUNS PROGRAM 7 AS A SUBROUTINE TO VERIFY THE
392                                 CORRECT TRANSFER OF DATA THROUGH THE INTERFACE AND PROGRAM 1
393                                 RUNS PROGRAM 11 AS A SUBROUTINE TO VERIFY THE CURSOR ADDRESSING FEATURES
394
395                             E.  REFER TO SECTION 6 ERRORS.  ERRORS DETECTED ARE LOGGED
396                                 ON THE CONSOLE DEVICE.
397
398                             EXECUTION TIME: - PROGRAM RUNS CONTINUOUSLY.
```

5. PROGRAM AND/OR OPERATOR ACTION

5.1 NORMAL HALTS

LOC CHLTA — COMMON HALT. THIS HALT OCCURS WHENEVER THE PROGRAM IS AWAITING USER INTERVENTION. THE DATA LIGHTS CONTAIN THE ADDRESS OF INSTRUCTION THAT GENERATED THE CALL TO THE COMMON HALT.

LOC RHLTA — END OF ROUTINE HALT. THIS HALT OCCURS AT THE END OF A TEST ROUTINE IF SR15 IS SET TO A 1. TO PROCEED, PRESS CONTINUE. PROGRAMS PRG0, PRG1, AND PRG2 USE THE ROUTINE END OPTION.

LOC PRGEND. — PROGRAM END HALT. THIS HALT NORMALLY OCCURS AT THE END OF PROGRAMS PRG0, PRG1 AND UNLESS THE LOOP PROGRAM OPTION IS SET. (SR10). THIS HALT WILL NOT OCCUR AT THE END OF PROG 0 OR PROG 1 UNDER MONITOR CONTROL OR IF THE CONSTANTS PER 7.4 HAVE BEEN ALTERED.

6. ERRORS

6.1 ERROR HALTS

LOC ERRLT — UNCONDITIONAL ERROR HALT. DATA LIGHTS CONTAIN ADDRESS OF INSTRUCTION THAT GENERATED THE ERROR CALL. REFER TO PROGRAM LISTING.

LOC CERRH — CONDITIONAL ERROR HALT. THIS CALL WILL ALWAYS OCCUR, UNLESS SR14 IS SET TO A 1 (SCOPE MODE) AND THE ERROR HAS OCCURRED AT LEAST ONCE. DATA LIGHTS CONTAIN ADDRESS OF INSTRUCTION THAT GENERATED ERROR CALL. REFER TO PROGRAM LISTING.

LOC DERRH — DATA ERROR HALT. OCCURS WHEN A PROGRAM OR ROUTINE CHECKING DATA FINDS THAT THE EXPECTED AND THE RECEIVED DATA DO NOT AGREE. THE LEFT HALF OF THE DATA LIGHTS CONTAIN THE EXPECTED 8 BIT DATA. THE RIGHT HALF CONTAINS THE RECEIVED 8 BIT DATA.

UNDER MONITOR LOAD AND WHEN CONSTANTS HAVE BEEN ALTERED PER 7.4 FOR GROUP TESTIN AN ERROR MESSAGE IS TYPED ON THE CONSOLE DEVICE INDICATING TYPE OF ERROR, ADDRES (CSR) OF DEVICE IN ERROR AND PROGRAM ADDRESS OF TEST. NO HALT OCCURS AND THE TE ADVANCES TO THE NEXT DEVICE. IMPROPER SR SETTING (NON ZERO) FOR MONITOR LOAD WILL GENERATE A TEST ABORTED MESSAGE

6.2 NON RECOVERABLE ERROR HALTS

A NON-RECOVERABLE ERROR HALT WILL OCCUR AT THE ADDRESSES LISTED BELOW IF THROUGH HARDWARE OR SOFTWARE FAILURE, PROGRAM CONTROL IS TRANSFERRED TO AN UNEXPECTED AREA BETWEEN 000000 AND 000176.

000002 RESERVED AREA
000006 ERROR TRAP
000012 RESERVED INSTRUCTION TRAP
000016 DEBUG TRAP
000022 IOT TRAP
000026 POWER FAIL TRAP
000044 THROUGH 000776 SYSTEM SOFTWARE AND INTERRUPT VECTOR AREA, EXCEPT FOR KL11 INTERRUPT VECTORS.

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

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

A NON-RECOVERABLE ERROR HALT FAILURE IS AN ABNORMAL CONDITION INDICATING A HARDWARE FAILURE, OR MOST UNLIKELY, A PROGRAM FAILURE. THIS PROGRAM ASSUMES THAT THE PROCESSOR IS IN OPERATING CONDITION IN ORDER TO TEST THE VTOS. ANY FURTHER STEPS TO DIAGNOSE A NON-RECOVERABLE ERROR ARE NOT WITHIN THE SCOPE OF THIS PROGRAM.

7. MISCELLANEOUS

7.1 SR OPTIONS

THE STANDARD SR OPTIONS ARE DESCRIBED HERE.

SR15 - HALT AT END OF ROUTINE. FOR THESE PROGRAMS CONSISTING OF A SET OF SEPARATE TEST ROUTINES, SR15 SET TO A 1 CAUSES THE PROGRAM TO HALT UPON COMPLETION OF THE ROUTINE CURRENTLY BEING EXECUTED. THREE POSSIBLE USES OF THIS OPTION ARE:

A. TO STEP THROUGH A PROGRAM ONE ROUTINE AT A TIME.

B. WHEN AN UNPREDICTED FAILURE HAS OCCURRED (BLOW UP, HANG UP), TO ADVANCE THROUGH THE PROGRAM ONE ROUTINE AT A TIME UNTIL THE FAILURE OCCURS. THE ROUTINE FOLLOWING THE LAST IDENTIFIED ROUTINE WOULD BE THE FAILING ROUTINE.

C. WHEN A PROGRAM IS IN EXECUTION, TO DETERMINE HOW FAR THE PROGRAM HAS PROGRESSED.

SR14 - SCOPE. THIS OPTION IS USED ONLY BY PRGO. THE OPTION CAUSES THE PROGRAM TO BYPASS ERROR HALTS, AND TO STAY IN THE FAILING ROUTINE. THIS OPTION WILL NOT BECOME ACTIVE UNTIL AN ERROR OCCURS. SR14 MUST BE ON BEFORE THE ERROR OCCURS, OR AT LEAST IT MUST BE SET BEFORE PRESSING CONTINUE AFTER AN ERROR HALT.

(7.2 CONT'D)

SR11 - INHIBIT ITERATION COUNT. THIS OPTION IS USED BY PRGO, PRG1, AND PRG3. THESE PROGRAMS CONSIST OF A SET OF ROUTINES EACH OF WHICH SPECIFIES THE NUMBER OF TIMES A TEST IS TO BE PERFORMED BY MEANS OF AN ITERATION COUNT. SETTING SR11 TO A 1 CAUSES THE PROGRAM TO DISREGARD THE ITERATION COUNT AND PERFORM THE TEST ONLY ONCE FOR EACH ROUTINE. TWO POSSIBLE USES OF THIS OPTION ARE:

A. QUICK PASS. THE USER MAY ELECT TO RUN THROUGH A PROGRAM QUICKLY TO FIND OUT IF ANY FAILURES SHOW IMMEDIATELY. A SUCCESSFUL QUICK PASS HOWEVER, DOES NOT GUARANTEE THAT THE SAME PROGRAM WILL RUN ERROR-FREE WHEN PERFORMING A NORMAL ITERATION PASS.

B. SKIP OVER FAILING ROUTINE. WHEN A ROUTINE HAS DETECTED A SOLID FAILURE, THE ERROR WILL BE REPORTED MANY TIMES. TO GO ON TO THE NEXT ROUTINE, THE USER CAN INHIBIT ITERATION. IT WILL BE NECESSARY TO CAUSE THE PROGRAM TO STOP AT THE END OF THE ROUTINE BY SETTING SR15 TO A 1. OTHERWISE THE PROGRAM WOULD QUICKLY RUN THROUGH THE NEXT ROUTINE(S) ALSO.

SR10 - LOOP PROGRAM. THIS OPTION IS USED BY PROGRAMS PRGO,PRG1, AND PRG4. SETTING SR10 TO A 1 CAUSES THE PROGRAM TO REPEAT ITSELF UPON COMPLETION, INSTEAD OF STOPPING AT PROGRAM END HALT.

SR9 - SELECT ROUTINE. THIS OPTION IS USED BY PROGRAMS PRGO, PRG1 AND PRG4. THE USER MAY ELECT TO RUN ONLY ONE SPECIFIC ROUTINE BY SETTING SR9 TO A 1, AND SR6 THROUGH SR0 TO THE NUMBER OF THE DESIRED ROUTINE. REFER TO THE INDIVIDUAL PROGRAM DESCRIPTION IN SECTION 8 TO OBTAIN THE ROUTINE NUMBER. THE ROUTINE NUMBER SELECTED MUST BE A VALID NUMBER, OR AN ERROR HALT WILL OCCUR. THE SELECT ROUTINE OPTION WILL BE HONORED BY THE PROGRAM UPON COMPLETION OF THE CURRENT ROUTINE, OR UPON STARTING THE PROGRAM.

SR8 - DISABLE STALL MODE AND RUN FULL SPEED. USED BY PROGRAM PRG10. THIS PROGRAM OPERATES NORMALLY IN STALL MODE (TESTS OR EXERCISES ARE NOT FULL SPEED, BUT RANDOM DURATION DELAYS ARE INTRODUCED). SETTING SR8 TO A 1 CAUSE THE PROGRAM TO PERFORM THEIR TESTS AT FULL SPEED.

7.3 TESTING VT05 AT NON-STANDARD ADDRESSES AND/OR VECTORS

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

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

LOCATION	FROM STANDARD	TO NON-STANDARD
CONADD	177560	RECEIVER CSR ADDRESS
CONVEC	000060	RECEIVER VECTOR ADDRESS

B. PROCEED TO USE PROGRAM, OR

C. USING STANDARD DUMP ROUTINES, DUMP OUT THE ENTIRE PROGRAM IN ABSOLUTE FORMAT, TO HAVE AN UPDATED OBJECT TAPE THAT REFLECTS YOUR SYSTEM, OR

D. DUMP OUT ONLY LOCATIONS 000204 THROUGH 000222, AND SPLICE THE TAPE TO THE END OF THE STANDARD OBJECT TAPE. THIS PROCEDURE WOULD REQUIRE THAT THE SHORT LENGTH OF TAPE BE LOADED IMMEDIATELY AFTER THE MAIN PROGRAM, IN ORDER TO OVERLAY LOCATIONS 000204 THROUGH 000222.

7.4 TESTING A CONTIGUOUS GROUP OF VT05'S

THIS PROGRAM WILL AUTOMATICALLY AND CONTINUOUSLY TEST THE CONSOLE DEVICE AND A CONTIGUOUS GROUP OF VT05'S PROVIDED CERTAIN CONSTANTS OF THE PROGRAM ARE CHANGED AS FOLLOWS:

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

LOCATION	FROM STANDARD	TO NON-STANDARD
000042	000000	000200
LOWADD	176500	FIRST ADDNL. VT05 RECEIVER CSR ADDRESS
LSTADD	000000	LAST + 10 VT05 RECEIVER CSR ADDRESS

8. DESCRIPTION

NOTE: THIS TEXT WAS ORIGINALLY WRITTEN TO DESCRIBE A TELETYPE. THEREFORE, INTERPRET READER/KYBD AS RECEIVER AND PUNCH AS TRANSMITTER.

8.1 PRG0 PROGRAM DESCRIPTION

PRG0 TESTS THE INPUT AND OUTPUT LOGIC IN ONE PROGRAM. THE PROGRAM CONSISTS OF 35 TEST ROUTINES NUMBERED FROM 00 TO 42(8).

RTN0 TESTS ABILITY TO REFERENCE THE READER/KYBD STATUS WORD (TKS) WITHOUT TRAPPING.

RTN1 TESTS ABILITY TO REFERENCE THE READER/KYBD BUFFER (TKB) WITHOUT TRAPPING.

RTN2 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH STATUS WORD (TPS) WITHOUT TRAPPING.

RTN3 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH BUFFER (TPB) WITHOUT TRAPPING.

RTN4 TESTS ABILITY TO SET AND CLEAR THE READER/KYBD ID BIT.

RTN5 CHECKS THAT READER /KYBD ID BIT CAN BE CLEARED WITH RESET INSTRUCTION.

RTN11 CHECKS THAT READER DONE BIT SETS NO LATER THAN 200 MSECS AFTER READER ENABLE.

RTN12 TESTS THAT READER/KYBD DONE BIT CAN BE READ RELIABLY.

RTN13 CHECKS THAT RESET INSTRUCTION CLEARS THE READER DONE BIT.

RTN14 CHECKS THAT REFERENCING READER BUFFER CLEARS DONE BIT.

RTN17 TESTS THAT READ BUFFER CAN BE READ RELIABLY.

RTN20 CHECKS THAT READER DONE BIT IS ABLE TO CAUSE AN INTERRUPT. IF THE INTERRUPT IS SERVICED, IT WILL HAVE OCCURRED AT CORRECT VECTOR.

RTN21 TESTS THAT READER DONE DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS AT THE SAME PRIORITY AS THE READER'S INTERRUPT REQUEST LEVEL.

RTN22 TESTS THAT READER DONE CAUSES INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY ONE LEVEL LOWER THAN THE READER'S INTERRUPT REQUEST LEVEL.

RTN23 CHECKS THAT READER DONE DOES NOT REINTERRUPT AFTER RTI INSTRUCTION WHEN DONE BIT IS LEFT SET.

(8.1 CONT'D)

RTN24 TESTS ABILITY TO SET AND CLEAR THE PUNCH ID BIT.
RTN25 CHECKS THAT PUNCH ID BIT CAN BE CLEARED WITH RESET INSTRUCTION.
RTN26 TESTS ABILITY TO SET AND CLEAR PUNCH MAINTENANCE BIT.
RTN27 CHECKS THAT RESET INSTRUCTION CLEARS THE MAINTENANCE BIT.
RTN30 TESTS THAT RESET SETS THE PUNCH READY BIT, AND THAT THE READY BIT CAN BE READ RELIABLY.
RTN31 TESTS THAT PUNCH READY IS CLEARED BY LOADING THE PUNCH BUFFER.
RTN32 TESTS THAT BYTE LOADING PUNCH BUFFER+1 DOES NOT CLEAR THE PUNCH READY BIT.
RTN33 CHECKS THAT THE PUNCH BECOMES READY NO LATER THAN 200 MSECS AFTER BUFFER LOAD.
RTN34 CHECKS THAT PUNCH READY BIT CAN CAUSE INTERRUPT. IF THE INTERRUPT IS SERVICED, IT WILL HAVE OCCURRED AT THE CORRECT VECTOR
RTN35 TESTS THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WHEN THE PROCESSOR IS AT A PRIORITY AS THE READER'S INTERRUPT REQUEST LEVEL.
RTN36 TESTS THAT PUNCH READY CAUSES AN INTERRUPT WHEN THE PROCESSOR IS AT PRIORITY ONE LEVEL LOWER THAN THE PUNCH INTERRUPT REQUEST LEVEL.
RTN40 CHECKS THAT PUNCH READY CAUSES AN INTERRUPT IMMEDIATELY UPON LOWERING PROCESSOR PRIORITY TO 0.
RTN41 CHECKS FOR CORRECT OPERATION OF WAIT INSTRUCTION. (REFER TO PROGRAM LISTING).
RTN42 TESTS THAT LOADING PUNCH BUFFER WITH MAINTENANCE BIT SET CAUSES READER DONE BIT TO SET NO LATER THAN 290 MSECS.
RTN43 IF MONITOR ADDRESS IN NONZERO (LOADED BY THE MONITOR OR GROUP TESTING DESIRED) THIS ROUTINE WILL LINK TO PROG. 7 WHICH VERIFIES THAT PROPER TRANSFERS CAN BE EXECUTED TO AND FROM THE BUFFERS

```


8.2     PRG1 PROGRAM DESCRIPTION
        ------------------------

        PRG1 EXERCISES THE DISPLAY FUNCTIONS. VERIFICATION OF DISPLAY
        OPERATION IS PERFORMED VISUALLY BY USER. THE PROGRAM CONTAINS
        29 ROUTINES NUMBERED FROM 00 TO 34(8).

        RTN0    CARRIAGE RETURN TEST. CHECKS THAT CARRIAGE CAN CORRECTLY
                RETURN TO PRINT POSITION 0 FROM EVERY OTHER PRINT POSITION.
                VISUAL DISPLAY IS A WEDGE.


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

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

        RTN2    RIGHT CURSOR TEST. CHECKS ABILITY OF THE VT05 TO
                SPACE CORRECTLY WITH THE "RIGHT CURSOR" CHARACTER
                THE TEST FIRST PRINTS REVERSE SLASHES (\) IN ALTERNATE
                PRINT POSITIONS, AND THEN FROM PRINT POSITION 0 TO EACH
                PRINT POSITION AND PRINTS A SLASH (/). THE TYPEOUT SHOULD
                LOOK AS FOLLOWS:

\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/





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

```
735
736
737                                     RTN4    TAB TEST.  CHECKS ABILITY OF THE DISPLAY TO TAB
738                                             POSITIONS 1 TO 9.  THE FIRST LINE TYPED MARKS THE TAB
739                                             POSITIONS.  SUBSEQUENT LINES INCREMENT BY ONE SPACE
740                                             EACH.  THE TYPEOUT LOOKS AS FOLLOWS:
741
742                                     /       /       /       /       /       /       /       /       /
743
744
745                                     /       /       /       /       /       /       /       / /
746
747
748                                     /       /       /       /       /       /       /       / /
749
750
751                                     /       /       /       /       /       /       /       / /
752
753
754                                     /       /       /       /       /       /       /       /   /
755
756
757                                     /       /       /       /       /       /       /       /       /
758
759
760                                     /       /       /       /       /       /       /       /       /
761
762
763                                     /       /       /       /       /       /       /       /       /
```

```
                        (8.2 CONT'D)

                RTN5    TYPES LINE OF CHARACTERS ABC
                RTN6    TYPES LINE OF CHARACTERS DEF
                RTN7    TYPES LINE OF CHARACTERS GHI
                RTN10   TYPES LINE OF CHARACTERS JKL
                RTN11   TYPES LINE OF CHARACTERS MNO
                RTN12   TYPES LINE OF CHARACTERS PQR
                RTN13   TYPES LINE OF CHARACTERS STU
                RTN14   TYPES LINE OF CHARACTERS VWX
                RTN15   TYPES LINE OF CHARACTERS YZ0
                RTN16   TYPES LINE OF CHARACTERS 123
                RTN17   TYPES LINE OF CHARACTERS 456
                RTN20   TYPES LINE OF CHARACTERS 789
                RTN21   TYPES LINE OF CHARACTERS !"#
                RTN22   TYPES LINE OF CHARACTERS $%&
                RTN23   TYPES LINE OF CHARACTERS '()
                RTN24   TYPES LINE OF CHARACTERS *+,
                RTN25   TYPES LINE OF CHARACTERS -./
                RTN26   TYPES LINE OF CHARACTERS :;<
                RTN27   TYPES LINE OF CHARACTERS =>?
                RTN30   TYPES LINE OF CHARACTERS @[\
                RTN31   TYPES LINE OF CHARACTERS ]↑ AND LEFT ARROW
                RTN32   TYPES 2 LINES OF ALL CHARACTERS. FIRST LINE IS TYPED AT
                        FULL SPEED, SECOND LINE IS TYPED WITH RANDOM STALLS.
                RTN33   TYPES 12 LINES OF ASR33 (001224=10) WORST CASE PATTERN.
                        EVERY OTHER LINE IS TYPED WITH RANDOM STALLS. THE ASR33
                        WORST CASE PATTERN IS '←W/W←
                RTN34   TYPES 12 LINES OF ASR35 (001224=11) WORST CASE PATTERN.
                        EVERY OTHER LINE IS TYPED WITH RANDOM STALLS. THE ASR35
                        WORST CASE PATTERN IS '[?C?[
                RTN35   IF MONITOR ADDRESS IS NONZERO (LOADED BY
                        THE MONITOR OR GROUP TESTING DESIRED) THIS ROUTINE
                        WILL LINK TO PROGRAM 11 WHICH EXERCISES
                        THE CURSOR ADDRESSING FEATURES.
```

8.3 PRG2 PROGRAM DESCRIPTION

PRG2 IS USED TO TEST THE TELETYPE KEYBOARD. THE PROGRAM CONTAINS 3 ROUTINES NUMBERED FROM 00 TO 02.

DEFAULT PROGRAM FOR SWITCHLESS PROCESSOR

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

RTN1 ECHO TEST. THE TEST ECHOES ONTO THE TELEPRINTER THE CHARACTER RECEIVED FROM THE KEYBOARD. WHEN THE TEST SENSES A RUBOUT CHARACTER THE TEST IS ENDED. THE TEST ENABLES THE USER TO DETERMINE IF ALL PRINTABLE CODES CAN BE SUCCESSFULLY SENT TO THE VT05 CONTROL. THE FOLLOWING SECTIONS (8.3.1, 8.3.2) DESCRIBE HOW THIS ROUTINE SHOULD BE USED TO TEST THE SPECIAL CHARACTERS.

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

8.3.1 CURSOR TEST

A. USING "SPACES" AND "LINE FEEDS" PUT THE LETTER "A" AT A KNOWN POSITION (5 "SPACES" AND 5 "LINE FEEDS")

B. PLACE A "B" AT ANOTHER KNOWN POSITION (10 MORE "SPACES" AND 5 MORE "LINE FEEDS")

C. HOME UP

D. USING "CURSOR RIGHT" AND CURSOR DOWN" POSITION THE CURSOR OVER THE "B". IT SHOULD REQUIRE EXACTLY 15 "CURSOR RIGHT"S AND 10 "CURSOR DOWN"S.

E. USING "CURSOR LEFT" AND "CURSOR UP" POSITION THE CURSOR OVER THE "A". IT SHOULD REQUIRE EXACTLY 5 "CURSOR-UP"S AND 10 "CURSOR LEFT"S.

8.3.2 ERASE TEST

A. FILL THE SCREEN WITH ANY CHARACTER AND RETURN CURSOR TO LEFT SIDE OF SCREEN.

B. TYPE EOL (ERASE LINE) AND "CURSOR UP".

C. REPEAT B 25 TIMES AND SCREEN SHOULD BE CLEAR.

D. FILL SCREEN WITH ANY CHARACTER.

E. "HOME UP" AND TYPE "EOF" (ERASE FIELD) AND SCREEN SHOULD BE CLEAR.

DESCRIPTION OF CURSOR CONTROL CODES AND SPECIAL FUNCTION CODES

TABLE 8-3

OCTAL CODE	CORRESPONDING KEY	DESCRIPTION
07	BELL	CAUSES A "BEEP" TO SOUND IN THE SPEAKER INSIDE THE DISPLAY.
12	LINE FEED	CAUSES THE CURSOR TO MOVE DOWN ONE LINE POSITION UNTIL THE CURSOR REACHES THE BOTTOM LINE. WHEN THE CURSOR IS ON THE BOTTOM LINE, THE CODE WILL CAUSE THE TEXT TO ROLL UP ONE LINE (TOP LINE IS LOST).
15	RETURN	CAUSES THE CURSOR TO RETURN TO THE FIRST CHARACTER POSITION ON THE SAME LINE. CARRIAGE RETURN/ LINE FEED WILL NOT AUTOMATICALLY OCCUR AFTER THE SEVENTY-SECOND CHARACTER IS WRITTEN.
10	BACK SPACE	CAUSES THE CURSOR TO MOVE BACK- WARD ONE CHARACTER SPACE. IF THE CURSOR IS AT CHARACTER POSITION ONE, THIS CODE HAS NO EFFECT.
37	ERASE EOF	CAUSES THE SCREEN TO BE ERASED FROM THE CURSOR POSITION TO THE END OF THE SCREEN.
36	ERASE EOL	CAUSES THE PORTION OF THE LINE FROM THE CURSOR POSITION TO THE END OF THAT LINE TO BE ERASED.
35	HOME UP	PLACES THE CURSOR IN THE FIRST CHARACTER POSITION OF THE FIRST LINE ON THE SCREEN.
34	HOME DOWN	PLACES THE CURSOR IN THE FIRST CHARACTER POSITION OF THE LAST LINE ON THE SCREEN.
32	CURSOR UP	MOVES THE CURSOR VERTICALLY UPWARD TO THE NEXT HIGHER LINE. IF THE CURSOR IS ON THE TOP LINE, THIS CODE HAS NO EFFECT.
13	CURSOR DOWN	MOVES THE CURSOR VERTICALLY DOWN- WARD ONE LINE. IF THE CURSOR

```
919                                                     IS ON THE BOTTOM LINE, THIS HAS
920                                                     NO EFFECT.
```

OCTAL CODE	CORRESPONDING KEY	DESCRIPTION
31	CURSOR LEFT	MOVES THE CURSOR HORIZONTALLY LEFT ONE POSITION. IF THE CURSOR IS AT CHARACTER POSITION ONE, THIS CODE HAS NO EFFECT.
30	CURSOR RIGHT	MOVES THE CURSOR HORIZONTALLY RIGHT ONE POSITION. IF THE CURSOR IS AT CHARACTER POSITION 72, THIS CODE HAS NO EFFECT.
40	SPACE	THE CHARACTER AT THE CURSOR POSITION WHEN THE SPACE COMMAND IS ISSUED WILL BE ERASED THE CURSOR MOVES HORIZONTALLY ONE CHARACTER POSITION TO THE RIGHT IF THE CURSOR IS IN CHARACTER POSITION 72, THAT CHARACTER AT THE 72ND POSITION IS ERASED, BIUT THE CURSOR WILL REMAIN AT THE 72ND POSITION.

THE CURSOR SWITCH AT THE REAR OF THE UNIT WILL CAUSE THE VTOS
TO IGNORE (NO RESPONSE ON THE SCREEN) THE FOLLOWING CODES

10 BACK SPACE
37 ERASE EOF
36 ERASE EOL
35 HOME UP
34 HOME DOWN
32 CURSOR UP
13 CORSOR DOWN
31 CURSOR LEFT
30 CURSOR RIGHT WHEN IN THE "OFF" POSITION.

8.8 PRG3 PROGRAM DESCRIPTION

PRG3 IS A PRINTER EXERCISER DESIGNED AS AN AID IN MAKING VT05 ADJUSTMENTS. THE PROGRAM PERMITS THE USER TO TYPE IN FIVE TEST CHARACTERS AND ONE FINAL CHARACTER THAT SIGNIFIES WHETHER FULL SPEED OR STALL OPERATION IS DESIRED. THE PROGRAM THEN TYPES LINES CONTAINING THE FIVE SELECTED CHARACTERS. WHEN THE USER WISHES TO CHANGE THE TEST CHARACTERS SR15 IS SET TO A 1. THE PROGRAM TERMINATES TYPING THE LINE BEFORE ACCEPTING NEW DATA.

THIS TEST CAN ALSO BE USED FOR ALIGNMENT BY FILLING THE SCREEN WITH E'S.

8.4 PRG4 PROGRAM DESCRIPTION

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

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

8.5 PRG5 PROGRAM DESCRIPTION

PRG5 IS USED AS AN AID IN ADJUSTING THE RECEIVER CLOCK. A SCOPE IS REQUIRED. THE PROGRAM PERFORMS THE FOLLOWING SEQUENCE:

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

8.6 PRG6 PROGRAM DESCRIPTION

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

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

8.7 PRG7 PROGRAM DESCRIPTION

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

A. INITIALIZE BINARY COUNT PATTERN.
B. SET PUNCH MAINTENANCE BIT.
C. LOAD PUNCH BUFFER WITH BINARY COUNT CHARACTER.
D. WHEN READER DONE BIT SETS, COMPARE BINARY CHARACTER WITH DATA IN READ BUFFER. HALT IF NOT SAME.
E. WAIT FOR PUNCH DONE BIT TO SET AND GO TO STEP C.
F. THIS PROGRAM IS EXECUTED AS A SUBROUTINE OF PROGRAM 0 UNDER MONITOR LOAD OR WHEN AUTOMATIC GROUP TESTING IS USED.

```
1037
1038
1039
1040
1041                                    8.10    PROGRAM 10 DESCRIPTION
1042                                            ----------------------
1043
1044                                            THE PURPOSE OF THIS TEST IS TO VERIFY THAT THE VT05 MEMORY HAS
1045                                            ROLL-UP CAPABILITIES.  THE TEST FUNCTIONS AS FOLLOWS:
1046
1047                                            A.  A LINE !, AND ITS COMPLEMENT   , IS DISPLAYED
1048                                            B.  THIS LINE IS FOLLOWED BY A LINE OF ITS EXACT COMPLEMENT
1049
1050                                            THIS PROCEDURE RUNS CONTINUOUSLY.  IF SWITCH 15 IS HELD UP
1051                                            MOMENTARILY THE ASCII CODE FOR THE CHARACTER IS INCREMENTED
1052                                            BY ONE.  BY UTILIZING SWITCH 15 IN THIS MANNER, PROGRAM 10 CAN
1053                                            TEST THE ROLL-UP CAPABILITY OF ALL CHARACTERS.
1054
1055                                    8.11    PROGRAM 11 DESCRIPTION
1056                                            ----------------------
1057
1058                                                    RANDOM NUMBERS ARE GENERATED AND USED AS X AND Y
1059                                            COORDINATES. A 1440 CHARACTER MESSAGE IS USED AS THE VIS-
1060                                            UAL DISPLAY, ONE PARTICULAR CHARACTER IS AVAILABLE FOR
1061                                            EACH CURSOR POSITION.
1062                                                    THE FIRST LOCATION OF THE STORED CHARACTERS (MIN-
1063                                            IMUM COORDINATES) IS USED AS A CONSTANT. A RANDOM Y COOR-
1064                                            DINATE (40 TO 63) IS GENERATED AND CONVERTED TO 0 TO 23.
1065                                            THIS LATER NUMBER IS THEN MULTIPLIED BY 110 WHICH PLACES
1066                                            US AT THE BEGINNING OF LINES 0 TO 19. AN X COORDINATE (40
1067                                            TO 147) IS NOW GENERATED AND CONVERTED TO 0 TO 107 AND ADD-
1068                                            ED TO THE Y COORDINATE, ADDRESS, AND THIS CONTROLS THE TYPE
1069                                            OF CHARACTER OUTPUT.
1070                                                    AFTER 1440 CHARACTERS ARE PRINTED A HALT WILL OCCUR
1071                                            TO ALLOW VISUAL INSPECTION. A RERUN IS POSSIBLE BY DEPRESS-
1072                                            ING THE CONTINUE KEY.
1073                                                    THIS PROGRAM IS EXECUTED AS A SUBROUTINE OF PROGRAM
1074                                            0 UNDER MONITOR LOAD OR WHEN AUTOMATIC GROUP TESTING IS
1075                                            USED.
1076                                    .ENDR
1077
1078
```

```
                                .TITLE KL11 TESTS FOR VT05 DISPLAY TERMINAL
                                ;MAINDEC 11-DZVTB-D
                                ;KL11  TESTS FOR VT05 DISPLAY TERMINAL
                                ;AUTHOR:         JIM CARRON
                                ;DATE:           JUNE, 1972
                                ;REVISED:        MARCH, 1973
                                ;COPYRIGHT 1971,1972,1973 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                                ;
                                ;PRG0- COMBINED INPUT-OUTPUT LOGIC TESTS.
                                ;PRG1- DISPLAY TEST.
                                ;PRG2- KEYBOARD TEST.
                                ;PRG3- PRINTER EXERCISER.
                                ;PRG4- PUNCH CLOCK ADJUSTMENT ROUTINE.
                                ;PRG5- READER CLOCK ADJUSTMENT ROUTINE.
                                ;PRG6- MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
                                ;PRG7- MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST.
                                ;PRG10- ROLE-UP TEST
                                ;PRG11-  CURSOR ADDRESS TEST

                                ;STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
                                ;
                                ;SR 15 - HALT AT END OF ROUTINE.
                                ;SR 14 - SCOPE.
                                ;SR 11 - INHIBIT ITERATION.
                                ;SR 10 - LOOP PROGRAM.
                                ;SR  9 - SELECT ROUTINE.
                                ;SR 8 - DISABLE STALL MODE AND RUN FULL SPEED.
                                ;SR 6 THROUGH SR 0 - NUMBER OF ROUTINE TO BE SELECTED.
                                        .ENABL  ABS
                                        .=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  000340                  PRTY7
000030  002776                  EMTINT                          ;EMT TRAP
000032  000340                  PRTY7
000034  003042                  TRPINT                          ;TRAP TRAP. SIMILAR TO EMT
000036  000340                  PRTY7
000040  000042                  .+2
000042  000000          MONITR: HALT                            ;TRAPPED FOR MONITOR LOAD
                                .=46
000046  002614                  $ENDAD
                                .REPT   176
                                .+2
                                HALT                            ;TRAPPED TO PREVIOUS ADDRESS.
                                .ENDR
```

```
1133
1134                                    ;EQUATE STATEMENTS
1135            177776                          CC=177776
1136            177776                          PSW=177776
1137            000240                          NOP=240
1138            000000                          OPEN=0
1139            100000                          HLTSW=BIT15             ;HALT SWITCH DEFINITION
1140            040000                          SCOPSW=BIT14            ;SCOPE SWITCH DEFINITION
1141            020000                          NPRTSW=BIT13            ;INHIBIT PRINT SWITCH DEFINITION
1142            010000                          NTRCSW=BIT12            ;INHIBIT TRACE SWITCH DEFINITION
1143            004000                          NITRSW=BIT11            ;INHIBIT ITERATION SWITCH DEFINITION
1144            002000                          LPRGSW=BIT10            ;LOOP PROGRAM SWITCH DEFINITION
1145            001000                          SRTSW=BIT9              ;SELECT ROUTINE SWITCH DEFINITION
1146            000400                          BYPMAN=BIT8             ;BYPASS MANUAL INTERVENTION DEFINITION.
1147            100000                          MANUAL=BIT15
1148            100000                          BIT15=100000
1149            040000                          BIT14=40000
1150            020000                          BIT13=20000
1151            010000                          BIT12=10000
1152            004000                          BIT11=4000
1153            002000                          BIT10=2000
1154            001000                          BIT9=1000
1155            000400                          BIT8=400
1156            000200                          BIT7=200
1157            000100                          BIT6=100
1158            000040                          BIT5=40
1159            000020                          BIT4=20
1160            000010                          BIT3=10
1161            000004                          BIT2=4
1162            000002                          BIT1=2
1163            000000                          BIT0=0
1164            005726                          POPSP=5726              ;POP THE STACK. SAME AS TST (6)+
1165            022626                          POPSP2=022626           ;POP STACK TWICE. SAME AS CMP (6)+,(6)+
1166            000340                          PRTY7=340               ;PRIORITY LEVEL DEFINITIONS
1167            000300                          PRTY6=300
1168            000240                          PRTY5=240
1169            000200                          PRTY4=200
1170            000140                          PRTY3=140
1171            000100                          PRTY2=100
1172            000040                          PRTY1=40
1173            000000                          PRTY0=0
1174            104000                          TYPE=EMT+0
1175            104001                          TYPES=EMT+1
1176            104002                          STALL=EMT+2
1177            104003                          ERROR=EMT+3
1178            104004                          DATCHK=EMT+4
1179            104005                          CHALT=EMT+5
1180            104006                          STRDRV=EMT+6
1181            104007                          STPCHV=EMT+7
1182            104010                          EHALT=EMT+10
1183            104011                          SRESET=EMT+11
1184            104012                          CHAIN=EMT+12
1185            104013                          CK33=EMT+13
```

```
                  104014                          CK35=EMT+14
                  104016                          TYPLN3=EMT+16
                  104017                          DATHLT=EMT+17
                  104020                          SAVREG=EMT+20
                  104021                          RSTREG=EMT+21
                  104022                          CHKASR=EMT+22
                  104400                          DELAY=TRAP+0
                  000007                          BELL=007
                  017621                          BLOCKA=DEND
                  017623                          BLOCK1=BLOCKA+2
                  017733                          BLOCKB=BLOCKA+112
                  017744                          BLKBB=BLOCKA+123
                  017735                          BLOCK2=BLOCKA+114
                  017746                          BLK2=BLOCKA+125
                  020045                          BLOCKC=BLOCKA+224
                  020056                          BLKCC=BLOCKA+235
                  000174                          .=174
        000174    000000                DISPREG:  0
        000176    000000                SWREG:    0
                  000200                          .=200
        000200    000167    001534                JMP       START           ;GO TO START OF PROGRAM.
                  001104                .=.+700                       ;GET CODE OUT OF VECTOR AREA
        001104    000000                SPBOT:    0                         ;BOTTOM OF STACK
        001106    177570                SR:       177570
        001110    177570                DISPLAY:177570
        001112    177560                CONADD: 177560                      ;CONSOLE LSR CSR POINTER
        001114    000060                CONVEC: 60                          ;CONSLE INTERRUPT VECTOR
        001116    177560                TKS:      177560                    ;LSR CSR
        001120    177562                TKB:      177562                    ;LSR BUFFER
        001122    177564                TPS:      177564                    ;LSP CSR
        001124    177566                TPB:      177566                    ;LSP BUFFER
        001126    000060                TKVTR:    60                        ;LSR INTERRUPT VECTOR
        001130    000200                TKLVL:    PRTY4                     ;LSR PRIORITY LEVEL
        001132    000064                TPVTR:    64                        ;LSP INTERRUPT VECTOR
        001134    000200                TPLVL:    PRTY4                     ;LSP PRIORITY LEVEL
        001136    000001                TTYTYP: 01                          ;TTY = KSR35
        001140    000000                PRGNUM: OPEN                        ;CONTAINS CURRENT PROGRAM#
        001142    000000                KSTART: OPEN                        ;CURRENT PROGRAM START ADDRESS.
        001144    000000                CURTST: OPEN                        ;CONTAINS ADDR OF CURRENT TEST.
        001146    000000                RTNNO:    OPEN                      ;CONTAINS CURRENT TEST #.
        001150    000000                NXTST:    OPEN                      ;CONTAINS ADDR OF NEXT TEST.
        001152    000000                ICTR:     OPEN                      ;CONTAINS CURRENT ITERATION COUNT
        001154    000000                SCOPTR: OPEN                        ;CONTAINS CURRENT SCOPE POINTER.
        001156    000000                PRGID:    OPEN                      ;CONTAINS PROGRAM INDICATORS
        001160    005712                PRGTAB: PRG0                        ;PRG0 START ADDRESS
        001162    010260                          PRG1                      ;PRG1 START ADDRESS
        001164    011704                          PRG2                      ;PRG2 START ADDRESS
        001166    012204                          PRG3                      ;PRG3 START ADDRESS
        001170    012370                          PRG4                      ;PRG4 START ADDRESS
        001172    012400                          PRG5                      ;PRG5 START ADDRESS
        001174    012470                          PRG6                      ;PRG6 START ADDRESS
        001176    012542                          PRG7                      ;PRG7 START ADDRESS
        001200    012654                          PRG10                     ;PRG10 START ADDRESS
        001202    013014                          PRG11                     ;PRG11 START ADDRESS
```

```
1242
1243  001204  004136  EMTTAB:  TYP          ;POINTER TO TYPEOUT ROUTINE
1244  001206  004462           TYPS         ;POINTER TO CHAINED MESSAGES ROUTINE
1245  001210  004560           STAL         ;POINTER TO RANDOM STALL ROUTINE
1246  001212  001536           ERR          ;POINTER TO ERROR ROUTINE
1247  001214  001436           DTCHK
1248  001216  001300           CHLT         ;COMMON HALT
1249  001220  003766           STLSRV
1250  001222  004016           STLSPV
1251  001224  001322           EHLT         ;POINTER TO ERROR HALT ROUTINE.
1252  001226  004046           SRSETT
1253  001230  002430           CHAINN
1254  001232  003206           CHK33
1255  001234  003222           CHK35
1256  001236  003220           CHK330
1257  001240  005430           TYPL3
1258  001242  001526           DTHLT
1259  001244  003562           SAVRG
1260  001246  003622           RSTRG
1261  001250  003240           CKASR
1262  001252  004514  TRPTAB:  DLY
1263  001254  000000  RCNT:    OPEN         ;CHARACTER COUNT
1264  001256  000000  CRBUF:   OPEN         ;HOLDS ONE CHARACTER FROM READER.
1265  001260  000000  CHR1:    OPEN
1266  001262  000000  CHR2:    OPEN
1267  001264  000000  CHR3:    OPEN
1268  001266  000000  ERCTR:   OPEN
1269  001270  000000  CTRA:    OPEN
1270  001272  000000  CTRB:    OPEN
1271  001274  000000  CTRC:    OPEN
1272  001276  000000  CTRD:    OPEN
```

```
1273
1274                                        ;COMMON HALT ROUTINE
1275  001300  022767  000000  176534  CHLT: CMP     #0,MONITR       ;MONITOR LOAD?
1276  001306  001004                      BNE     CHLTR           ;YES - EXIT
1277  001310  011600                      MOV     @%6,%0          ;DEVELOP ADDRESS OF CALLER
1278  001312  162700  000002              SUB     #2,%0
1279  001316  000000                      HALT                    ;HALT, ADDRESS OF CALL INSTRUCTION
1280  001320  000002              CHLTR:  RTI                     ;IN DATA LIGHTS
1281                                        ;UNCONDITIONAL ERROR HALT ROUTINE
1282  001322  022767  000000  176512  EHLT: CMP     #0,MONITR       ;MONITOR LOAD?
1283  001330  001435                      BEQ     EHLTA           ;NO - HLT.
1284  001332  104011                      SRESET
1285  001334  004767  001724          EHLTB: JSR     %7,CONIT        ;SET UP CONSOLE POINTERS
1286  001340  016700  002206              MOV     CURADD,%0       ;SET UP DEVICE NUMBER
1287  001344  162700  000010              SUB     #10,%0          ;OF DISPLAY IN ERROR.
1288  001350  012767  016424  003514      MOV     #ERRORA,A1ST    ;SET ERROR DEVICE ADDRESS
1289  001356  004767  000332              JSR     %7,ACNVE        ;CONVERT OCTAL TO ASCII
1290  001362  112777  000067  003502      MOVB    #67,@A1ST
1291  001370  011600                      MOV     @%6,%0          ;TRANSFER PROGRAM ERROR
1292  001372  162700  000002              SUB     #2,%0           ;ADDRESS TO MESSAGE
1293  001376  012767  016453  003466      MOV     #ERRORB,A1ST
1294  001404  004767  000304              JSR     %7,ACNVE
1295  001410  104000                      TYPE                    ;TYPE ERROR MESSAGE
1296  001412  016362                      ERRORM
1297  001414  004767  001722              JSR     %7,RSTART
1298  001420  000167  000566              JMP     NSTART
1299  001424  011600          EHLTA:  MOV     @%6,%0          ;DEVELOP ADDRESS OF CALLER
1300  001426  162700  000002              SUB     #2,%0
1301  001432  000000          ERRH:   HALT                    ;HALT ADDRESS OF ERROR CALL
1302  001434  000002                      RTI                     ;IN DATA LIGHTS.
1303                                        ;DATA CHECK ROUTINE.
1304  001436  126767  177614  177613  DTCHK: CMPB    CRBUF,CRBUF+1   ;COMPARE EXPECTED AND RECEIVED
1305  001444  001433                      BEQ     DTCHKA          ;CHARS. BRANCH IF SAME.
1306  001446  022767  000000  176366      CMP     #0,MONITR       ;MONITOR LOAD?
1307  001454  001424                      BEQ     DTHLT           ;NO GO TO NORMAL HALT
1308  001456  004767  001602              JSR     %7,CONIT        ;SET UP CONSOLE POINTERS
1309  001462  016700  002064              MOV     CURADD,%0       ;SET UP DEVICE NUMBER OF
1310  001466  162700  000010              SUB     #10,%0          ;DISPLAY IN ERROR
1311  001472  012767  016566  003372      MOV     #DERRA,A1ST     ;SET ERROR MESSAGE ADDRESS
1312  001500  004767  000210              JSR     %7,ACNVE        ;CONVERT OCTAL TO ASCII
1313  001504  112777  000067  003360      MOVB    #67,@A1ST       ;MOST SIG.DIGIT = 7
1314  001512  104000                      TYPE                    ;TYPE ERROR MESSAGE
1315  001514  016541                      DERR
1316  001516  004767  001620              JSR     %7,RSTART
1317  001522  000167  000464              JMP     NSTART
1318  001526  016700  177524          DTHLT: MOV     CRBUF,%0        ;MOVE S/B AND WAS CHARS TO R0.
1319  001532  000000          DERRH:  HALT                    ;DATA ERROR HALT. GOOD CHAR IN
1320                                                          ;DATA LIGHTS LEFT. BAD CHAR IN DATA
1321  001534  000002          DTCHKA: RTI                     ;LIGHTS RIGHT. EXIT.
1322                                        ;CONDITIONAL ERROR HALT.
1323  001536  032777  040000  177342  ERR:  BIT     #SCOPSW,@SR     ;CHECK SCOPE SWITCH.
1324  001544  001404                      BEQ     ERRA            ;BRANCH IF NO SCOPE DESIRED.
1325  001546  005767  177404              TST     PRGID           ;SCOPING WANTED. FIRST ERROR?
1326  001552  100001                      BPL     ERRA            ;NO SCOPE IF FIRST ERROR.
1327  001554  000002                      RTI                     ;SCOPE EXIT.
1328  001556  052767  100000  177372  ERRA: BIS     #BIT15,PRGID    ;SET ERROR INDICATOR.
```

```
1329  001564  022767  000000  176250          CMP     #0,MONITR       ;MONITOR LOAD?
1330  001572  001434                          BEQ     ERRB            ;NO - GO TO NORMAL HALT
1331  001574  104011                          SRESET
1332  001576  004767  001462                  JSR     %7,CONIT        ;SET UP CONSOLE POINTERS
1333  001602  016700  001744                  MOV     CURADD,%0       ;SET UP DEVICE NUMBER
1334  001606  162700  000010                  SUB     #10,%0          ;OF DISPLAY IN ERROR
1335  001612  012767  016503  003252          MOV     #CERRA,A1ST     ;SET ERROR MESSAGE ADDRESS
1336  001620  004767  000070                  JSR     %7,ACNVE        ;CONVERT OCTAL TO ASCII
1337  001624  112777  000067  003240          MOVB    #67,@A1ST       ;MOST SIG DIGIT = 7
1338  001632  012767  016532  003232          MOV     #CERRB,A1ST     ;TRANSFER PROGRAM ERROR
1339  001640  011600                          MOV     @%6,%0          ;ADDRESS TO MESSAGE
1340  001642  162700  000002                  SUB     #2,%0
1341  001646  004767  000042                  JSR     %7,ACNVE        ;CONVERT OCTAL TO ASCII
1342  001652  104000                          TYPE                    ;TYPE ERROR MESSAGE
1343  001654  016463                          CERR
1344  001656  004767  001460                  JSR     %7,RSTART
1345  001662  000553                          BR      NSTART
1346  001664  011600                ERRB:     MOV     @%6,%0          ;DEVELOP ADDRESS OF
1347  001666  162700  000002                  SUB     #2,%0           ;CALLER
1348  001672  000000                CERRH:    HALT                    ;ERROR HALT
1349  001674  000002                          RTI                     ;EXIT
1350                                  ;ROUTINE END HALT SUBROUTINE.
1351  001676  005777  177204        $HALT:    TST     @$R             ;CHECK HALT SWITCH.
1352  001702  100003                          BPL     SHLTA           ;BRANCH IF NO HALT DESIRED.
1353  001704  116700  177236                  MOVB    RTNNO,%0        ;CURRENT TEST # TO R0.
1354  001710  000000                RHLTA:    HALT                    ;ROUTINE END HALT.
1355  001712  000207                SHLTA:    RTS     %7              ;EXIT.
1356  001714  016701  003152        ACNVE:    MOV     A1ST,%1         ;TRANSFER ADDRESS TO R1
1357  001720  062701  000006                  ADD     #6,%1           ;ADD 6 TO REF LSD
1358  001724  012702  000006                  MOV     #6,%2           ;SET ROUTINE CTR TO 6
1359  001730  010067  003144                  MOV     %0,ACNVX        ;TRANSFER NUMBER TO CONVERT ROUTINE REG
1360  001734  000167  003156                  JMP     ACNVM           ;GO TO CONVERT ROUTINE
```

```
                                                ;ROUTINE FOR MONITOR LOAD
001740  012706  001104          START:  MOV     #SPBOT,%6       ;SET BOTTOM OF STACK
001744  012767  003066  176052          MOV     #PFAIL,24       ;SET POWER FAIL POINTER
001752  012767  177570  177126          MOV     #177570,SR
001760  012767  177570  177122          MOV     #177570,DISPLAY
001766  013746  000006                  MOV     @#6,-(%6)       ;SAVE ERROR VECTOR
001772  013746  000004                  MOV     @#4,-(%6)
001776  012767  002012  176000          MOV     #1$,4           ;SET UP TIME OUT VECTOR
002004  005777  177076                  TST     @SR             ;TRY TO REFERENCE HARDWARE SWR
002010  000407                          BR      2$              ;BRANCH IF NO TIME OUT TRAP OCCURS
002012  012767  000176  177066  1$:     MOV     #SWREG,SR       ;POINT TO SOFTWARE SWR
002020  012767  000174  177062          MOV     #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REG
002026  022626                          CMP     (%6)+,(%6)+     ;RESTORE STACK
002030  012637  000004          2$:     MOV     (%6)+,@#4       ;RESTORE ERROR VECTOR
002034  012637  000006                  MOV     (%6)+,@#6
002040  016767  001510  001504          MOV     LOWADD,CURADD   ;INITIALIZE POINTER FOR START OF MULTIPLES
002046  012767  002104  175730          MOV     #END,4          ;INITIALIZE TIME OUT TRAP
002054  012767  000300  001476          MOV     #300,CURVEC     ;INITIALIZE VECTOR POINTER
002062  012700  174000                  MOV     #174000,%0      ;ADDRESS OF FIRST DC11 TO R0
002066  005710                  B:      TST     @%0             ;IS THIS DEVICE EQUPPED?
002070  062767  000010  001462          ADD     #10,CURVEC      ;NO ADD #10 TO VECTOR AND
002076  062700  000010                  ADD     #10,%0          ;ADDRESS
002102  000771                          BR      B
002104  022626                  END:    POPSP2
002106  016702  001442                  MOV     LOWADD,%2
002112  162702  176500                  SUB     #176500,%2      ;SUBTRACT GROUP OFFSET
002116  060267  001436                  ADD     %2,CURVEC       ;ADD GROUP BIAS TO VECTOR POINTER.
002122  004767  001136                  JSR     %7,CONIT        ;RESTORE POINTERS & VECTORS TO CONSOLE
002126  005767  175710                  TST     MONITR
002132  001007                          BNE     1$              ;IF SOFTWARE SWR
002134  022767  000176  176744          CMP     #176,SR         ;PRESENT AND NOT IN
002142  001003                          BNE     1$              ;MONITOR MODE DEFAULT
002144  052777  000002  176734          BIS     #2,@SR          ;TO PROGRAM #2
002152  012767  000000  001370  1$:     MOV     #0,STNUM        ;RESET PROGRAM LOADER
002160  022767  000000  175654          CMP     #0,MONITR       ;IS THIS A MONITOR LOAD?
002166  001427                          BEQ     STARTO          ;NO - GO TO REGULAR START
002170  022777  000000  176710          CMP     #0,@SR          ;ANY PANEL SW SET?
002176  001405                          BEQ     NSTART          ;NO - GO TO NEW START
002200  104000                          TYPE                    ;TYPE ABORT MESSAGE
002202  016575                          ABORT
002204  000005                          RESET
002206  000167  011254                  JMP     LOGICAL         ;RETURN TO MONITOR.
002212  012706  001104          NSTART: MOV     #SPBOT,%6       ;SET BOTTOM OF SP STACK
002216  005067  175554                  CLR     PSW
002222  012767  000006  175554          MOV     #6,MACHER
002230  005067  176712                  CLR     RTNNO
002234  016700  001310                  MOV     STNUM,%0        ;TRANSFER PROGRAM NUMBER TO R0
002240  005267  001304                  INC     STNUM
002244  000422                          BR      CRTA
002246  012706  001104          STARTO: MOV     #SPBOT,%6       ;SET BOTTOM OF SP STACK.
002252  005067  175520                  CLR     PSW
002256  012767  000006  175520          MOV     #6,MACHER
002264  005067  176656                  CLR     RTNNO
002270  017700  176612                  MOV     @SR,%0          ;(SR) TO R0
002274  042700  177760                  BIC     #177760,%0      ;LIMIT (SR) TO BITS 3-0
```

```
1417  002300  020027  000014            CMP     %0,#14          ;COMPARE (SR) TO PROGRAM LIMIT
1418  002304  101402                    BLOS    CRTA            ;VALID PROGRAM NUMBER?
1419  002306  104010            INCPRG: EHALT                   ;NO. INCORRECT PRG NUMBER
1420  002310  000613                    BR      START           ;START OVER.
1421  002312  005067  176640    CRTA:   CLR     PRGID
1422  002316  010067  176616            MOV     %0,PRGNUM       ;SAVE PROGRAM NUMBER AT PRGNUM
1423  002322  006100                    ROL     %0              ;ROX2
1424  002324  000170  001160            JMP     @PRGTAB(0)      ;GO TO SELECTED PROGRAM.
1425  002330  104005            SRSET:  CHALT                   ;SET SR OPTIONS DESIRED
1426  002332  016767  176604  176610 GETRDY: MOV     KSTART,NXTST    ;ADDR OF 1ST ROUTINE TO NXTST
1427  002340  000167  000372            JMP     CLEAN           ;GO CLEAN UP.
1428  002344  004767  000262    GTRDYA: JSR     %7,FORWD        ;ROLL FORWARD TO "NEXT" ROUTINE.
1429  002350  032777  001000  176530 GTRDYB: BIT     #SRTSW,@SR      ;CHECK FOR SELECT ROUTINE SWITCH
1430  002356  001003                    BNE     GTRDYC          ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
1431  002360  004767  000324            JSR     %7,GOTST        ;GO RUN CURRENT ROUTINE.
1432  002364  000460                    BR      CHNB            ;NO GO. MANUAL RTN BYPASSED.
1433  002366  017700  176514    GTRDYC: MOV     @SR,%0          ;(SR) TO R0
1434  002372  042700  177600            BIC     #177600,%0      ;MASK UNDESIRED BITS
1435  002376  126700  176544            CMPB    RTNNO,%0        ;COMPARE RTNNO TO (R0)
1436  002402  001004                    BNE     GTRDYD          ;BRANCH IF ROUTINE NOT FOUND YET.
1437  002404  004767  000300            JSR     %7,GOTST        ;GO RUN ROUTINE.
1438  002410  104010                    EHALT                   ;NO GO. MANUAL RTN SELECTED BYPASSED.
1439  002412  000747                    BR      GETRDY
1440  002414  022767  177777  176526 GTRDYD: CMP     #-1,NXTST       ;NO. CHECK FOR LAST ROUTINE.
1441  002422  001350                    BNE     GTRDYA          ;LAST ROUTINE?
1442  002424  104010            INCRTN: EHALT                   ;YES. INCORRECT ROUTINE SELECTED.
1443  002426  000741                    BR      GETRDY          ;START OVER.
1444  002430  005767  176522    CHAINN: TST     PRGID           ;TEST ERROR BIT IN PRGID.
1445  002434  100013                    BPL     CHNA            ;BRANCH IF ERROR BIT NOT SET.
1446  002436  032777  040000  176442    BIT     #SCOPSW,@SR     ;ERROR BIT SET. CHECK FOR SCOPE OPTION.
1447  002444  001407                    BEQ     CHNA            ;SCOPE SWITCH SET IN SR?
1448  002446  022767  177777  176500    CMP     #-1,SCOPTR      ;YES. CHECK SCOPE ENTRY POINTER
1449  002454  001403                    BEQ     CHNA            ;BRANCH IF SCOPE ENTRY IS -1.
1450  002456  017716  176472            MOV     @SCOPTR,@%6     ;SET UP TO GO SCOPING
1451  002462  000002                    RTI                     ;GO TO SCOPE ENTRY.
1452  002464  042767  100000  176464 CHNA:   BIC     #BIT15,PRGID    ;CLEAR ERROR BIT IN PRGID.
1453  002472  032777  004000  176406        BIT     #NITRSW,@SR     ;TEST INHIBIT ITERATION SWITCH
1454  002500  001004                    BNE     CHNAA           ;INHIBIT ITERATION?
1455  002502  005367  176444            DEC     ICTR            ;NO
1456  002506  001401                    BEQ     CHNAA           ;COUNT 0?
1457  002510  000002                    RTI                     ;NO. RETURN TO TEST ROUTINE
1458  002512  022626            CHNAA:  POPSP2                  ;POP STACK TWICE
1459  002514  104011                    SRESET
1460  002516  004767  001140            JSR     %7,RETPVT       ;RESTORE VECTOR TABLES
1461  002522  004767  177150            JSR     %7,SHALT        ;GO HALT IF HALT SWITCH IS SET
```

```
 1462
 1463  002526  032777  001000  176352  CHNB:   BIT     #SRTSW,@SR      ;CHECK SELECT ROUTINE SWITCH
 1464  002534  001276                          BNE     GETRDY          ;SELECT ROUTINE SWITCH SET?
 1465  002536  022767  177777  176404          CMP     #-1,NXTST       ;NO.
 1466  002544  001275                          BNE     GTRDYA-4        ;LAST TEST?
 1467  002546  032777  002000  176332          BIT     #LPRGSW,@SR     ;YES. TEST LOOP PROGRAM SWITCH.
 1468  002554  001266                          BNE     GETRDY          ;LOOP PROGRAM?
 1469  002556  022767  000000  175256          CMP     #0,MONITR       ;MONITOR LOAD?
 1470  002564  001417                          BEQ     PRGEND          ;YES - GO TO HALT
 1471  002566  026727  000756  000001          CMP     STNUM,#1        ;PROGRAM #1 COMPLETED?
 1472  002574  003606                          BLE     NSTART          ;YES GO TO NEW START
 1473  002576  004767  000540                  JSR     %7,RSTART       ;GO TO RESTART
 1474  002602  000603                          BR      NSTART
 1475  002604  013700  000042                  MOV     @#42,%0
 1476  002610  001405                          BEQ     PRGEND
 1477  002612  000005                          RESET
 1478  002614  004710                  $ENDAD: JSR     %7,(%0)
 1479  002616  000240                          NOP
 1480  002620  000240                          NOP
 1481  002622  000240                          NOP
 1482  002624  000167  177502          PRGEND: JMP     GETRDY
 1483  002630  000736                          BR      CHNB
 1484  002632  016705  176312          FORWD:  MOV     NXTST,%5        ;ADDR OF NEXT ROUTINE TO R5.
 1485  002636  012567  176304                  MOV     (5)+,RTNNO      ;GET NEXT ROUTINE NUMBER.
 1486  002642  012567  176302                  MOV     (5)+,NXTST      ;GET ADDR OF NEXT "NEXT" ROUTINE.
 1487  002646  105767  176304                  TSTB    PRGID           ;CHECK IF PROGRAM SCOPE AND I COUNT
 1488  002652  100407                          BMI     FORWDB          ;PARAMETERS. BRANCH IF NOT.
 1489  002654  012567  176272                  MOV     (5)+,ICTR       ;GET ITERATION COUNT.
 1490  002660  012567  176270                  MOV     (5)+,SCOPTR     ;GET SCOPE LOOP ENTRY POINTER.
 1491  002664  010567  176254          FORWDA: MOV     %5,CURTST       ;ADDR OF NOW CURRENT TEST TO CURTST.
 1492  002670  000207                          RTS     %7              ;EXIT FORWD SUBROUTINE.
 1493  002672  012767  177777  176254  FORWDB: MOV     #-1,SCOPTR      ;FORCE "NO SCOPE"
 1494  002700  012767  000001  176244          MOV     #1,ICTR         ;FORCE I COUNT OF 1
 1495  002706  000766                          BR      FORWDA
 1496  002710  005767  176232          GOTST:  TST     RTNNO           ;CHECK FOR MANUAL RTN.
 1497  002714  100005                          BPL     GOTSTA          ;BRANCH IF NOT MANUAL RTN.
 1498  002716  032777  000400  176162          BIT     #BYPMAN,@SR     ;MANUAL RTN. BYPASS IT?
 1499  002724  001401                          BEQ     GOTSTA          ;NO. RUN IT.
 1500  002726  000207                          RTS     %7              ;BYPASS MANUAL ROUTINE.
 1501  002730  005726                  GOTSTA: POPSP
 1502  002732  000177  176206                  JMP     @CURTST         ;GO RUN TEST.
 1503  002736  012767  001233  001166  CLEAN:  MOV     #1233,RP1       ;RESET RANDOM NOS.
 1504  002744  012767  005622  001162          MOV     #5622,RP2
 1505  002752  104011                          SRESET
 1506  002754  012767  000006  175022          MOV     #6,MACHER       ;RESET MACHER TRAP.
 1507  002762  005067  175010                  CLR     PSW
 1508  002766  012706  001104                  MOV     #SPBOT,%6       ;SET UP BOTTOM OF STACK.
 1509  002772  000167  177346                  JMP     GTRDYA
 1510  002776  011646                  EMTINT: MOV     @%6,-(6)        ;GET SAVED PC.
 1511  003000  162716  000002                  SUB     #2,@%6          ;DECREMENT PC BY 2.
 1512  003004  017616  000000                  MOV     @(6),@%6
 1513  003010  121627  000022                  CMPB    @%6,#22         ;CHECK THAT CALL IS
 1514  003014  101402                          BLOS    EMTA            ;WITHIN LIMITS.
 1515  003016  000000                          HALT                    ;CALL NOT WITHIN LIMITS.
 1516  003020  000776                          BR      .-2
 1517  003022  006116                  EMTA:   ROL     @%6             ;EMT ARG X 2.
```

```
1518  003024  042716  177001                BIC     #177001,@%6     ;REMOVE 7 MSB.
1519  003030  062716  001204                ADD     #EMTTAB,@%6     ;FORM EMT RTN ADDR.
1520  003034  017616  000000                MOV     @(6),@%6
1521  003040  000136                        JMP     @(6)+           ;GO TO EMT ROUTINE.
1522  003042  011646                TRPINT: MOV     @%6,-(6)        ;GET SAVED PC.
1523  003044  162716  000002                SUB     #2,@%6          ;DECREMENT PC BY 2.
1524  003050  017616  000000                MOV     @(6),@%6
1525  003054  121627  000000                CMPB    @%6,#0          ;CHECK THAT EMT
1526  003060  101442                        BLOS    TRPA            ;IS WITHIN LIMITS.
1527  003062  000000                        HALT                    ;TRAP CALL NOT IN LIMIT.
1528  003064  000776                        BR      .-2
1529                                ;ENTER HERE FOR POWER FAIL
1530  003066  010046                PFAIL:  MOV     %0,-(6)         ;SAVE REGISTER OR STACK
1531  003070  010146                        MOV     %1,-(6)         ;WHEN POWERING DOWN
1532  003072  010246                        MOV     %2,-(6)
1533  003074  010346                        MOV     %3,-(6)
1534  003076  010446                        MOV     %4,-(6)
1535  003100  010546                        MOV     %5,-(6)
1536  003102  016746  174716                MOV     24,-(6)
1537  003106  017746  175774                MOV     @SR,-(6)
1538  003112  010667  000010                MOV     %6,SAVR6        ;STORE STACK POSITION
1539  003116  012767  003130  174700        MOV     #RESTART,24
1540  003124  000000                        HALT                    ;HALT ON POWER DOWN NORMAL
1541  003126  000000                SAVR6:  0                       ;STACK IS SAVED HERE
1542  003130  016706  177772        RESTART:        MOV     SAVR6,%6        ;RESTORE REGISTER OFF STACK
1543  003134  012677  175746                MOV     (6)+,@SR
1544  003140  012667  174660                MOV     (6)+,24         ;WHEN POWERING UP
1545  003144  012605                        MOV     (6)+,%5
1546  003146  012604                        MOV     (6)+,%4
1547  003150  012603                        MOV     (6)+,%3
1548  003152  012602                        MOV     (6)+,%2
1549  003154  012601                        MOV     (6)+,%1
1550  003156  012600                        MOV     (6)+,%0
1551  003160  000167  176554                JMP     START
1552  003164  000002                        RTI                     ;RETURN TO MAIN LINE
```

```
1553
1554  003166  006116                  TRPA:   ROL     @%6             ;TRAP ARG X 2.
1555  003170  042716  177001                  BIC     #177001,@%6     ;REMOVE 7 MSB.
1556  003174  062716  001252                  ADD     #TRPTAB,@%6     ;FORM TRAP RTN ADDR.
1557  003200  017616  000000                  MOV     @(6),@%6
1558  003204  000136                          JMP     @(6)+           ;GO TO TRAP ROUTINE.
1559  003206  005767  175724          CHK33:  TST     TTYTYP          ;CHECK FOR 33.
1560  003212  001002                          BNE     .+6             ;BRANCH IF NOT 33.
1561  003214  062716  000002                  ADD     #2,@%6          ;+2 TO EXIT POINTER
1562  003220  000002                  CHK330: RTI                     ;EXIT
1563  003222  022767  000001  175706  CHK35:  CMP     #1,TTYTYP       ;CHECK FOR 35.
1564  003230  001002                          BNE     .+6             ;BRANCH IF NOT 35.
1565  003232  062716  000002                  ADD     #2,@%6          ;+2 TO EXIT POINTER
1566  003236  000002                          RTI                     ;EXIT
1567  003240  032767  000010  175670  CKASR:  BIT     #BIT3,TTYTYP    ;CHECK FOR ASR TTY.
1568  003246  001001                          BNE     .+4             ;BRANCH IF NOT ASR.
1569  003250  000002                          RTI                     ;ASR. EXIT.
1570  003252  022626                          POPSP2                  ;POP STACK TWICE.
1571  003254  012767  000001  175670          MOV     #1,ICTR         ;FORCE I COUNT TO A 1.
1572  003262  104012                          CHAIN                   ;CHAIN TO BYPASS ROUTINE.
1573                          ;SUBROUTINE TO RESTORE POINTERS AND VECTORS FOR CONSOLE OUTPUT.
1574  003264  016700  175622          CONIT:  MOV     CONADD,%0       ;CONSOLE ADDRESS TO R0
1575  003270  010067  175622                  MOV     %0,TKS          ;SET LSR CSR TO CONSOLE
1576  003274  005720                          TST     (0)+            ;ADD 2 TO R0
1577  003276  010067  175616                  MOV     %0,TKB          ;SET LSR BUFFER TO CONSOLE
1578  003302  005720                          TST     (0)+            ;ADD 2 TO R0
1579  003304  010067  175612                  MOV     %0,TPS          ;SET LSP CSR TO CONSOLE
1580  003310  005720                          TST     (0)+            ;ADD 2 TO R0
1581  003312  010067  175606                  MOV     %0,TPB          ;SET LSP BUFFER TO CONSOLE
1582  003316  016767  175572  175602          MOV     CONVEC,TKVTR    ;SET LSR INTERRUPT VECTOR TO CONSOLE
1583  003324  016767  175564  175600          MOV     CONVEC,TPVTR    ;SET LSP INTERRUPT VECTOR TO CONSOLE
1584  003332  062767  000004  175572          ADD     #4,TPVTR
1585  003340  000207                          RTS     %7
1586                          ;SUBROUTINE FOR ADVANCING TEST.
1587  003342  012767  000000  000200  RSTART: MOV     #0,STNUM        ;RESET PROGRAM LOADER
1588  003350  022777  000000  175530          CMP     #0,@SR          ;ANY PR SWITCH SET?
1589  003356  001407                          BEQ     RSTA            ;NO - CONTINUE
1590  003360  004767  177700                  JSR     %7,CONIT        ;RESTORE POINTER TO CONSOLE
1591  003364  104000                          TYPE                    ;TYPE ABORT MESSAGE
1592  003366  016575                          ABORT
1593  003370  000005                          RESET
1594  003372  000167  010070                  JMP     LOGICAL         ;EXIT TO MONITOR
1595  003376  004767  000260          RSTA:   JSR     %7,RETPVT       ;RESET INTERRUPT VECTORS
1596  003402  016700  000144                  MOV     CURADD,%0       ;NEXT DEVICE ADD TO R0
1597  003406  026767  000140  000142          CMP     CURADD,LSTADD   ;IS THIS LAST ADDRESS
1598  003414  001442                          BEQ     RSTB            ;YES - GO TO EXIT
1599  003416  062767  000010  000126          ADD     #10,CURADD      ;COMPUTE ADDRESS FOR NEXT TEST.
1600  003424  010067  175466                  MOV     %0,TKS          ;NEW LSR CSR
1601  003430  062700  000002                  ADD     #2,%0
1602  003434  010067  175460                  MOV     %0,TKB          ;NEW LSR BUFFER
1603  003440  062700  000002                  ADD     #2,%0
1604  003444  010067  175452                  MOV     %0,TPS          ;NEW LSP CSR
1605  003450  062700  000002                  ADD     #2,%0
1606  003454  010067  175444                  MOV     %0,TPB          ;NEW LSP BUFFER
1607  003460  016700  000074                  MOV     CURVEC,%0       ;NEXT VECTOR ADD TO R0
1608  003464  062767  000010  000066          ADD     #10,CURVEC      ;COMPUTE ADDRESS FOR NEXT VECTOR
```

```
1609  003472  010067  175430                  MOV     %0,TKVTR        ;NEW LSR INTERRUPT VECTOR
1610  003476  062700  000004                  ADD     #4,%0
1611  003502  010067  175424                  MOV     %0,TPVTR        ;NEW LSP INTERRUPT VECTOR
1612  003506  012767  003522  174270          MOV     #RSTB,4         ;SETUP TRAP EXIT
1613  003514  005777  175376                  TST     @TKS            ;IS THIS DEVICE EQUIPPED?
1614  003520  000412                          BR      RSTC            ;EXIT.
1615  003522  016767  000024  000026  RSTB:   MOV     CURADD,LSTADD   ;SETUP LAST ADDRESS
1616  003530  004767  177530                  JSR     %7,CONIT        ;RESTORE POINTER TO CONSOLE
1617  003534  104000                          TYPE                    ;TYPE DONE MESSAGE
1618  003536  016340                          DONE
1619  003540  000005                          RESET
1620  003542  000167  007720                  JMP     LOGICAL         ;EXIT TO MONITOR
1621  003546  000207                  RSTC:   RTS     %7
1622  003550  000000                  STNUM:  OPEN
1623  003552  000000                  CURADD: OPEN
1624  003554  176500                  LOWADD: 176500
1625  003556  000000                  LSTADD: OPEN
1626  003560  000000                  CURVEC: OPEN
1627                                  ;SAVE REGS 0 TO 4 SUBROUTINE.
1628  003562  012667  000030          SAVRG:  MOV     (6)+,SVRPC      ;SAVE PC AND PSW.
1629  003566  012667  000026                  MOV     (6)+,SVRPSW
1630  003572  010446                          MOV     %4,-(6)         ;SAVE REGS 0 - 4
1631  003574  010346                          MOV     %3,-(6)         ;IN STACK.
1632  003576  010246                          MOV     %2,-(6)
1633  003600  010146                          MOV     %1,-(6)
1634  003602  010046                          MOV     %0,-(6)
1635  003604  016746  000010                  MOV     SVRPSW,-(6)     ;RESTORE PC AND PSW.
1636  003610  016746  000002                  MOV     SVRPC,-(6)
1637  003614  000002                          RTI                     ;EXIT.
1638  003616  000000                  SVRPC:  OPEN
1639  003620  000000                  SVRPSW: OPEN
1640                                  ;RESTORE REGS 0 TO 4 SUBROUTINE.
1641  003622  012667  000030          RSTRG:  MOV     (6)+,RSTPC      ;SAVE PC AND PSW.
1642  003626  012667  000026                  MOV     (6)+,RSTPSW
1643  003632  012600                          MOV     (6)+,%0         ;RESTORE REGS 0 - 4
1644  003634  012601                          MOV     (6)+,%1         ;FROM STACK.
1645  003636  012602                          MOV     (6)+,%2
1646  003640  012603                          MOV     (6)+,%3
1647  003642  012604                          MOV     (6)+,%4
1648  003644  016746  000010                  MOV     RSTPSW,-(6)     ;RESTORE PC AND PSW.
1649  003650  016746  000002                  MOV     RSTPC,-(6)
1650  003654  000002                          RTI                     ;EXIT
1651  003656  000000                  RSTPC:  OPEN
1652  003660  000000                  RSTPSW: OPEN
1653                                  ;SUBROUTINE TO RESET INTERRUPT VECTORS
1654  003662  016701  175240          RETPVT: MOV     TKVTR,%1        ;PLACE CURRENT TRAP VECTOR
1655  003666  010100                          MOV     %1,%0           ;ADDRESS IN R1, AND R1 TO R0.
1656  003670  062700  000002                  ADD     #2,%0           ;ADD #2 TO R0
1657  003674  010021                          MOV     %0,(1)+         ;STORE .+2 AT TKVTR
1658  003676  012721  000000                  MOV     #0,(1)+         ;STORE HALT AT TKVTR+2
1659  003702  062700  000004                  ADD     #4,%0
1660  003706  010021                          MOV     %0,(1)+         ;STORE .+2 AT TPVTR
1661  003710  012711  000000                  MOV     #0,@%1          ;STORE HALT AT TPVTR+2
1662  003714  000207                          RTS     %7
```

```
1663
1664                                        ;ROUTINE TO FETCH A CHARACTER
1665  003716  012767  000310  000040  AREAD:  MOV     #200.,BRCTR     ;SET UP DELAY COUNT.
1666  003724  052777  000004  175170          BIS     #4,@TPS         ;SET MAINTENANCE BIT
1667  003732  005077  175166                  CLR     @TPB            ;LOAD PUNCH BUFFER
1668  003736  105777  175154          ARDA:   TSTB    @TKS            ;CHECK DONE BIT.
1669  003742  100407                          BMI     ARDB            ;BRANCH IF DONE.
1670  003744  104400                          DELAY                   ;DELAY 1 MILLISECOND.
1671  003746  000001                          1
1672  003750  005367  000010                  DEC     BRCTR           ;TIME UP?
1673  003754  001370                          BNE     ARDA            ;BRANCH IF TIME NOT UP YET.
1674  003756  104010                          EHALT                   ;ERROR. NO RESPONSE FROM READER.
1675  003760  000756                          BR      AREAD           ;TRY AGAIN.
1676  003762  000207                  ARDB:   RTS     %7              ;EXIT
1677  003764  000000                  BRCTR:  OPEN
1678                                        ;ROUTINE TO SET LSR INTERRUPT VECTOR AND PRIORITY
1679  003766  017667  000000  000012  STLSRV: MOV     @(6),STPRA+2    ;MOVE VECTOR ADDR TO STPRA+2
1680  003774  062716  000002                  ADD     #2,@%6          ;SET UP EXIT
1681  004000  016701  175122                  MOV     TKVTR,%1
1682  004004  012721  000000          STPRA:  MOV     #OPEN,(1)+      ;SET VECTOR ADDRESS
1683  004010  016721  175114                  MOV     TKLVL,(1)+      ;SET PRIORITY
1684  004014  000002                          RTI                     ;EXIT
1685                                        ;ROUTINE TO SET LSP INTERRUPT VECTOR AND PRIORITY.
1686  004016  017667  000000  000012  STLSPV: MOV     @(6),STPPA+2    ;MOVE VECTOR ADDR TO STPPA+2
1687  004024  062716  000002                  ADD     #2,@%6          ;SET UP EXIT
1688  004030  016701  175076                  MOV     TPVTR,%1
1689  004034  012721  000000          STPPA:  MOV     #OPEN,(1)+      ;SET VECTOR ADDRESS.
1690  004040  016721  175070                  MOV     TPLVL,(1)+      ;SET PRIORITY
1691  004044  000002                          RTI                     ;EXIT.
1692                                        ;ROUTINE TO ISSUE RESET.
1693  004046  012700  052525          SRSETT: MOV     #52525,%0               ;DATA TO R0.
1694  004052  005100                          COM     %0              ;COMPLEMENT (R0).
1695  004054  010067  177770                  MOV     %0,SRSETT+2     ;(R0) TO SRSETT+2.
1696  004060  000005                          RESET                   ;ISSUE RESET. (R0) IS
1697  004062  000002                          RTI                     ;DISPLAYED. EXIT.
1698                                        ;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
1699  004064  016700  000042          RNGEN:  MOV     RP1,%0
1700  004070  006100                          ROL     %0
1701  004072  006100                          ROL     %0
1702  004074  066700  000034                  ADD     RP2,%0
1703  004100  010067  000026                  MOV     %0,RP1
1704  004104  006100                          ROL     %0
1705  004106  006100                          ROL     %0
1706  004110  066700  000020                  ADD     RP2,%0
1707  004114  006100                          ROL     %0
1708  004116  006100                          ROL     %0
1709  004120  010067  000010                  MOV     %0,RP2
1710  004124  016700  000002                  MOV     RP1,%0
1711  004130  000207                          RTS     %7              ;EXIT. NUMBER IN R0
1712  004132  001233                  RP1:    1233
1713  004134  005622                  RP2:    5622
```

```
 1714
 1715  004136  011600                        ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
 1716  004140  062716  000002          TYP:  MOV   @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
 1717  004144  011000                        ADD   #2,@%6          ;SET UP EXIT.
 1718  004146  112067  000306                MOV   @%0,%0          ;ADDRESS OF MESSAGE TO R0.
 1719  004152  122767  000100  000300  TYPA: MOVB  (0)+,TYPDAT     ;GET CHARACTER
 1720  004160  001013                        CMPB  #100,TYPDAT     ;CHECK FOR"@"CHARACTER
 1721  004162  112767  000000  000270        BNE   TYPC            ;BRANCH IF NOT"@".
 1722  004170  004767  000046                MOVB  #0,TYPDAT       ;NULL TO CLEAR DOUBLE BUFFER
 1723  004174  112767  000000  000256        JSR   %7,OUTTYP
 1724  004202  004767  000034                MOVB  #0,TYPDAT
 1725  004206  000002                        JSR   %7,OUTTYP
 1726  004210  122767  000045  000242  TYPC: RTI                   ;TERMINATOR CHAR. DONE. EXIT.
 1727  004216  001505                        CMPB  #45,TYPDAT      ;CHECK FOR"%".
 1728  004220  122767  000043  000232        BEQ   TYPF            ;BRANCH IF"%"
 1729  004226  001506                        CMPB  #43,TYPDAT      ;NOT"%".CHECK FOR"#".
 1730  004230  004767  000006                BEQ   TYPG            ;BRANCH IF "#"
 1731  004234  000744                        JSR   %7,OUTTYP       ;TYPE CHAR IN TYPDAT
 1732                                        BR    TYPA
                                    ;SUBROUTINE TO INSERT NULLS AND PRINT CHARACTER.  USES REGISTER
 1733                               ;LOCATION TYPDAT REFERENCED IN THE OUTPUT MESSAGE SUBROUTINE
 1734                               ;AND LINKS TO TYPDAT REGISTER IN THE LSPCH SUBROUTINE.
 1735  004236  110067  000216          OUT:  MOVB  %0,TYPDAT       ;MOVE LSPCH CHAR TO TYPDAT.
 1736  004242  122767  000012  000210  OUTTYP: CMPB  #12,TYPDAT     ;CHECK FOR LINE FEED.
 1737  004250  001437                        BEQ   OUTNUL          ;GO TO NULL ROUTINE
 1738  004252  122767  000013  000200        CMPB  #13,TYPDAT      ;CHECK FOR CURSOR DWN
 1739  004260  001433                        BEQ   OUTNUL          ;GO TO NULL ROUTINE
 1740  004262  122767  000035  000170        CMPB  #35,TYPDAT      ;CHECK FOR CURSOR HOME
 1741  004270  001427                        BEQ   OUTNUL          ;GO TO NULL ROUTINE
 1742  004272  122767  000032  000160        CMPB  #32,TYPDAT      ;CHECK FOR CURSOR UP
 1743  004300  001423                        BEQ   OUTNUL          ;GO TO NULL ROUTINE
 1744  004302  122767  000037  000150        CMPB  #37,TYPDAT      ;CHECK FOR ERASE SCREEN
 1745  004310  001417                        BEQ   OUTNUL          ;GO TO NULL ROUTINE
 1746  004312  122767  000016  000140        CMPB  #16,TYPDAT      ;CHECK FOR CAD OPERATION
 1747  004320  001407                        BEQ   OUTNCT          ;GO TO CAD NULL ROUTINE
 1748  004322  026727  000102  000004        CMP   NULL,#4         ;NULL CTR 4 OR GREATER?
 1749  004330  003412                        BLE   OUTCAR          ;IF NOT GREATER GO TO OUTPUT
 1750  004332  005367  000072                DEC   NULL            ;DEC NULL REGISTER
 1751  004336  000407                        BR    OUTCAR          ;GO TO OUTPUT.
 1752  004340  012767  000005  000062  OUTNCT: MOV  #5,NULL         ;SET NULL CTR TO 6
 1753  004346  000403                        BR    OUTCAR          ;GO TO OUTPUT
 1754  004350  012767  000004  000052  OUTNUL: MOV  #4,NULL         ;SET NULL CTR TO 4.
 1755  004356  116777  000076  174540  OUTCAR: MOVB TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
 1756  004364  105777  174532                TSTB  @TPS            ;WAIT FOR DONE FLAG.
 1757  004370  100375                        BPL   .-4
 1758  004372  026727  000032  000004        CMP   NULL,#4         ;IS COUNTER> 4
 1759  004400  003012                        BGT   OUTEX           ;IF GREATER EXIT
 1760  004402  022767  000000  000020        CMP   #0,NULL         ;IS NULL CTR = 0
 1761  004410  001406                        BEQ   OUTEX           ;IF ZERO EXIT
 1762  004412  005367  000012                DEC   NULL            ;DEC NULL CTR
 1763  004416  112767  000000  000034        MOVB  #0,TYPDAT       ;TRANSFER ZERO TO OUTPUT
 1764  004424  000754                        BR    OUTCAR
 1765  004426  000207          OUTEX:  RTS   %7                    ;EXIT
 1766  004430  000000          NULL:   OPEN
 1767  004432  112767  000015  000020  TYPF: MOVB  #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
 1768  004440  004767  177576                JSR   %7,OUTTYP       ;GO TYPE CHAR.
 1769  004444  112767  000012  000006  TYPG: MOVB  #12,TYPDAT      ;MOVE LF CODE TO TYPDAT.
```

```
 1770  004452  004767  177564                JSR     %7,OUTTYP       ;GO TYPE CHAR.
 1771  004456  000633                        BR      TYPA
 1772  004460  000000            TYPDAT: OPEN
 1773                            ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
 1774  004462  011600            TYPS:   MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
 1775  004464  062716  000002            ADD     #2,@%6          ;UPDATE TO NEXT MESSAGE ADDRESS
 1776  004470  011067  000014            MOV     @%0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
 1777  004474  022767  177777  000006    CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
 1778  004502  001001                    BNE     TYPSA           ;BRANCH IF NOT TERMINATOR.
 1779  004504  000002                    RTI                     ;TERMINATOR, EXIT
 1780  004506  104000            TYPSA:  TYPE                    ;CALL ON TYP SUB TO TYPE MESSAGE
 1781  004510  000000            TYPSB:  OPEN                    ;ADDRESS OF MESSAGE GOES HERE
 1782  004512  000763                    BR      TYPS            ;GO PROCESS NEXT MESSAGE
```

```
1783
1784                                        ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
1785  004514  011667  000036                DLY:    MOV     @%6,DLCNT       ;GET DELAY COUNT ADDRESS.
1786  004520  062716  000002                        ADD     #2,@%6          ;SET UP EXIT ADDRESS
1787  004524  017746  000026                        MOV     @DLCNT,-(6)     ;DELAY COUNT TO STACK
1788  004530  005067  173242                        CLR     PSW             ;SET PRIORITY 0
1789  004534  012746  002260                DLYA:   MOV     #2260,-(6)      ;1 MSEC COUNT TO STACK
1790  004540  005316                        DLYB:   DEC     @%6             ;DECREMENT 1 MSEC COUNT
1791  004542  001376                                BNE     DLYB            ;BRANCH IF NOT 0.
1792  004544  005726                                POPSP                   ;ZERO. UNCOVER MSECS. COUNT.
1793  004546  005316                                DEC     @%6             ;DECREMENT IT
1794  004550  001371                                BNE     DLYA            ;BR IF NOT DONE DELAYING
1795  004552  005726                                POPSP                   ;DONE
1796  004554  000002                                RTI                     ;EXIT.
1797  004556  000000                DLCNT:  OPEN                            ;CONTAINS MILLISECONDS COUNT ADDRESS.
1798                                      ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
1799                                      ;DETERMINED BY CONTENTS OF LOC STLMSK.
1800  004560  032767  040000  174370  STAL:   BIT     #BIT14,PRGID    ;TEST FOR STALLS ALLOWED.
1801  004566  001001                        BNE     STALAA          ;ALLOWED.
1802  004570  000002                        RTI                     ;NOT ALLOWED.
1803  004572  004767  177266          STALAA: JSR     %7,RNGEN        ;GO GET RANDOM NUMBER.
1804  004576  046700  000014                BIC     STLMSK,%0       ;# IN R0. APPLY STALL MASK.
1805  004602  001404                        BEQ     STALB           ;BRANCH IF RESULT IS 0.
1806  004604  010067  000002                MOV     %0,STALA
1807  004610  104400                        DELAY                   ;DELAY
1808  004612  000000                STALA:  OPEN                    ;DELAY COUNT
1809  004614  000002                STALB:  RTI                     ;DONE. EXIT.
1810  004616  000000                STLMSK: OPEN                    ;STALL MASK.
1811                                      ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
1812  004620  004767  177240          GRCNT:  JSR     %7,RNGEN        ;GET RANDOM NUMBER
1813  004624  046700  000010                BIC     RCMSK,%0        ;APPLY MASK
1814  004630  001773                        BEQ     GRCNT           ;TRY AGAIN IF RESULT 0
1815  004632  010067  000004                MOV     %0,RNCNT        ;COUNT TO RNCNT
1816  004636  000207                        RTS     %7              ;EXIT.
1817  004640  000000                RCMSK:  OPEN                    ;RANDOM CHARACTER MASK.
1818  004642  000000                RNCNT:  OPEN                    ;RANDOM CHARACTER COUNT.
```

```
 1819                                           ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
 1820  004644  012767  177777  000014  INBIN:  MOV     #-1,RIND        ;SET ALL VARIABLES
 1821  004652  004567  000300                  JSR     %5,BMOVE        ;TO MINUS 1.
 1822  004656  004666                          RIND
 1823  004660  004667                          RIND+1
 1824  004662  000013                          11.
 1825  004664  000207                          RTS     %7              ;EXIT
 1826  004666  000000          RIND:   OPEN
 1827  004670  000000          PTO:    OPEN
 1828  004672  000000          PT1:    OPEN
 1829  004674  000000          PIND:   OPEN
 1830  004676  000000          PTOP:   OPEN
 1831  004700  000000          PT1P:   OPEN
```

```
1832
1833                                        ;SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN R0
1834  004702  016767  177762  177762  GTBIN:  MOV     PT0,PT1         ;PREVIOUS BIN CHAR TO PT1
1835  004710  005167  177756                  COM     PT1
1836  004714  005167  177746                  COM     RIND
1837  004720  001002                          BNE     .+6
1838  004722  005267  177744                  INC     PT1
1839  004726  042767  177400  177736          BIC     #177400,PT1     ;MASK TO 8 BITS
1840  004734  016767  177732  177726          MOV     PT1,PT0         ;SAVE BIN CHAR IN PT0
1841  004742  016700  177724                  MOV     PT1,%0          ;BIN CHAR TO R0.
1842  004746  000207                          RTS     %7              ;EXIT.
1843  004750  016767  177722  177722  GTBINP: MOV     PT0P,PT1P       ;PREVIOUS BIN CHAR TO PT1P
1844  004756  005167  177716                  COM     PT1P
1845  004762  005167  177706                  COM     PIND
1846  004766  001002                          BNE     .+6
1847  004770  005267  177704                  INC     PT1P
1848  004774  042767  177400  177676          BIC     #177400,PT1P    ;MASK TO 8 BITS.
1849  005002  016767  177672  177666          MOV     PT1P,PT0P       ;SAVE BIN CHAR IN PT0P.
1850  005010  016701  177664                  MOV     PT1P,%1         ;BIN CHAR TO R1.
1851  005014  000207                          RTS     %7              ;EXIT.
1852                                        ;OCTAL TO ASCII CONVERT ROUTINES
1853  005016  012500                  ACNV6:  MOV     (5)+,%0         ;CONVERT TO 6 ASCII. GET OCTAL ADDRESS
1854  005020  012567  000012                  MOV     (5)+,ACNVB      ;GET ASCII ADDRESS
1855  005024  004767  000052                  JSR     %7,ACNV         ;CONVERT TO ASCII
1856  005030  004567  000122                  JSR     %5,BMOVE        ;MOVE 6 CHARS TO ASCII ADDRESS
1857  005034  005072                          A1ST
1858  005036  000000                  ACNVB:  OPEN
1859  005040  000006                          6
1860  005042  000205                          RTS     %5              ;EXIT
1861  005044  012500                  ACNV4:  MOV     (5)+,%0         ;CONVERT TO 4 ASCII. GET OCTAL ADDRESS
1862  005046  012567  000012                  MOV     (5)+,ACNVC      ;GET ASCII ADDRESS
1863  005052  004767  000024                  JSR     %7,ACNV         ;CONVERT TO ASCII
1864  005056  004567  000074                  JSR     %5,BMOVE        ;MOVE 4 CHARS TO ASCII ADDRESS.
1865  005062  005074                          A1ST+2
1866  005064  000000                  ACNVC:  OPEN
1867  005066  000004                          4
1868  005070  000205                          RTS     %5              ;EXIT
1869  005072  000000                  A1ST:   OPEN
1870  005074  000000                          OPEN
1871  005076  000000                          OPEN
1872  005100  000000                  ACNVX:  OPEN
1873  005102  012701  005100          ACNV:   MOV     #A1ST+6,%1      ;ADDR TO STORE ASCII TO R1
1874  005106  012702  000006                  MOV     #6,%2           ;6 TO R2
1875  005112  011067  177762                  MOV     @%0,ACNVX       ;OCTAL WORD TO ACNVX
1876  005116  016703  177756          ACNVM:  MOV     ACNVX,%3
1877  005122  042703  177770                  BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT OCTAL #
1878  005126  062703  000060                  ADD     #60,%3          ;ADD 60 TO CONVERT TO ASCII
1879  005132  110341                          MOVB    %3,-(1)         ;STORE ASCII BYTE
1880  005134  006067  177740                  ROR     ACNVX           ;MOVE NEXT OCTAL DIGIT TO LEAST
1881  005140  006067  177734                  ROR     ACNVX           ;SIGNIFICANT POSITION
1882  005144  006067  177730                  ROR     ACNVX
1883  005150  005302                          DEC     %2              ;DONE 6 TIMES?
1884  005152  001361                          BNE     ACNVM           ;NO. REPEAT.
1885  005154  000207                          RTS     %7              ;YES. EXIT.
```

```
1886
1887                                    ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1888  005156  104020                    BMOVE:  SAVREG                  ;SAVE REGS.
1889  005160  012501                            MOV     (5)+,%1         ;GET"FROM"ADDRESS
1890  005162  012502                            MOV     (5)+,%2         ;GET"TO"ADDRESS
1891  005164  012503                            MOV     (5)+,%3         ;GET COUNT
1892  005166  112122                    BMOVA:  MOVB    (1)+,(2)+       ;MOVE BYTE
1893  005170  005303                            DEC     %3              ;DECREMENT COUNT
1894  005172  001375                            BNE     BMOVA           ;BRANCH IF NOT DONE.
1895  005174  104021                            RSTREG                  ;RESTORE REGS.
1896  005176  000205                            RTS     %5              ;DONE EXIT
1897                                    ;SUBROUTINE TO CHECK FOR PUNCH READY.
1898  005200  105777  173716            CPRDY:  TSTB    @TPS            ;TEST FOR READY BIT.
1899  005204  100001                            BPL     CPRDYA          ;BRANCH IF READY NOT SET.
1900  005206  000207                            RTS     %7              ;OK. EXIT.
1901  005210  104010                    CPRDYA: EHALT                   ;NOT READY. HALT.
1902  005212  000772                            BR      CPRDY
1903                                    ;SUBROUTINE TO PUNCH ON LSP CHARACTER IN REG 0.
1904  005214  004767  177760            LSPCH:  JSR     %7,CPRDY        ;GO CHECK FOR PUNCH READY
1905  005220  004767  177012                    JSR     %7,OUT          ;GO TO OUTPUT ROUTINE.
1906  005224  005000                            CLR     %0
1907  005226  000207                            RTS     %7              ;DONE. EXIT.
1908                                    ;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
1909  005230  012700  017614            BDCNV:  MOV     #DECVAL,%0      ;SET UP ADDR TO STORE DECIMAL ASCII IN R0
1910  005234  013501                            MOV     @(5)+,%1        ;BINARY VALUE TO R1.
1911  005236  012702  005336                    MOV     #ADTENP,%2      ;ADDR OF TEN POWER STRING TO R2.
1912  005242  012767  000005  000060            MOV     #5,CNVCTR       ;SET UP FOR 5 POWER CONVERSIONS.
1913  005250  012267  000060            BDCNVA: MOV     (2)+,TENPWR     ;MOVE POWER OF TEN VALUE TO TENPWR.
1914  005254  004767  000010                    JSR     %7,SUBTEN       ;PERFORM CONVERSION
1915  005260  005367  000044                    DEC     CNVCTR          ;DONE 5 CONVERSIONS?
1916  005264  001371                            BNE     BDCNVA          ;BRANCH IF NOT YET 5.
1917  005266  000205                            RTS     %5              ;YES, EXIT.
1918  005270  005067  000036            SUBTEN: CLR     DIGIT           ;CLEAR DIGIT
1919  005274  166701  000034            SUBTNA: SUB     TENPWR,%1       ;SUBTRACT TEN POWER FROM BINARY VALUE.
1920  005300  103403                            BCS     SUBTNB          ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
1921  005302  005267  000024                    INC     DIGIT
1922  005306  000772                            BR      SUBTNA
1923  005310  066701  000020            SUBTNB: ADD     TENPWR,%1       ;RESTORE SUBTRACTED VALUE.
1924  005314  062767  000060  000010            ADD     #60,DIGIT       ;CONVERT (DIGIT) TO ASCII
1925  005322  116720  000004                    MOVB    DIGIT,(0)+      ;MOVE ASCII CHAR TO DECVAL FIELD.
1926  005326  000207                            RTS     %7              ;EXIT.
1927  005330  000000                    CNVCTR: OPEN
1928  005332  000000                    DIGIT:  OPEN
1929  005334  000000                    TENPWR: OPEN
1930  005336  023420                    ADTENP: 10000.
1931  005340  001750                            1000.
1932  005342  000144                            100.
1933  005344  000012                            10.
1934  005346  000001                            1
```

```
1935
1936                                        ;SUBROUTINE TO TYPE A LINE OF CHARACTERS
1937  005350  012767  000112  000050  TYPLN:  MOV     #74.,TCTR       ;72 TO CHAR COUNT +CR,LF
1938  005356  012704  017621          TYPLA:  MOV     #BLOCKA,%4      ;SET LINE ADDRESS IN R4.
1939  005362  104002                  TYPLB:  STALL                   ;STALL IF ALLOWED.
1940  005364  112400                          MOVB    (4)+,%0         ;GET CHARACTER
1941  005366  004767  177622                  JSR     %7,LSPCH        ;GO OUTPUT CHARACTER.
1942  005372  005367  000030                  DEC     TCTR            ;DONE?
1943  005376  001371                          BNE     TYPLB           ;BRANCH IF NOT DONE.
1944  005400  112767  000000  177052  TYPCLR: MOVB    #0,TYPDAT
1945  005406  004767  176630                  JSR     %7,OUTTYP
1946  005412  112767  000000  177040          MOVB    #0,TYPDAT
1947  005420  004767  176616                  JSR     %7,OUTTYP
1948  005424  000207                          RTS     %7              ;DONE. EXIT
1949  005426  000000                  TCTR:   OPEN
1950                                        ;SUBROUTINE TO TYPE LINE OF 3 CHARACTERS
1951  005430  011667  000016          TYPL3:  MOV     @%6,TPL3A       ;DEVELOP AND SET ADDRESS OF
1952  005434  017767  000012  000010          MOV     @TPL3A,TPL3A    ;DATA IN TPL3A.
1953  005442  062716  000002                  ADD     #2,@%6          ;SET UP EXIT.
1954  005446  004567  000034                  JSR     %5,FBF3         ;FILL BUFFER WITH 3 CHARACTERS
1955  005452  000000                  TPL3A:  OPEN
1956  005454  042767  040000  173474          BIC     #BIT14,PRGID    ;DISABLE STALLS.
1957  005462  004767  177662                  JSR     %7,TYPLN        ;GO TYPE LINE OF CHARACTERS.
1958  005466  000002                          RTI                     ;EXIT.
1959  005470  112767  000015  012123  STBF:   MOVB    #15,BLOCKA      ;SUB TO SET UP BUFFER AREA.
1960  005476  112767  000012  012116          MOVB    #12,BLOCKA+1
1961  005504  000207                          RTS     %7              ;EXIT
1962                                        ;SUBROUTINE TO FILL CHARACTER BUFFER WITH 3 CHARACTERS.
1963  005506  012567  000004          FBF3:   MOV     (5)+,FBF3A
1964  005512  004567  177440                  JSR     %5,BMOVE        ;MOVE 3 CHARS TO BUFFER.
1965  005516  000000                  FBF3A:  OPEN
1966  005520  017623                          BLOCK1
1967  005522  000003                          3
1968  005524  004567  177426          FBF3B:  JSR     %5,BMOVE        ;FILL 72 CHARACTERS BUFFER
1969  005530  017623                          BLOCK1                  ;WITH 3 CHARACTERS
1970  005532  017626                          BLOCK1+3
1971  005534  000105                          69.
1972  005536  004567  177414                  JSR     %5,BMOVE
1973  005542  017623                          BLOCK1
1974  005544  017735                          BLOCK2
1975  005546  000110                          72.
1976  005550  000205                          RTS     %5              ;EXIT
```

```
1977
1978                                    ;SUBROUTINE TO FILL BUFFER WITH ALL CHARACTERS
1979  005552  004567  177400    FBALL:  JSR     %5,BMOVE        ;FILL 72 CHAR BUFFER WITH
1980  005556  016646                    A                       ;ALL CHARACTERS.
1981  005560  017623                    BLOCK1
1982  005562  000077                    63.
1983  005564  004567  177366            JSR     %5,BMOVE
1984  005570  016646                    A
1985  005572  017722                    BLOCK1+63.
1986  005574  000011                    9.
1987  005576  004567  177354            JSR     %5,BMOVE
1988  005602  017623                    BLOCK1
1989  005604  017735                    BLOCK2
1990  005606  000110                    72.
1991  005610  000207                    RTS     %7              ;EXIT.
1992                                    ;SUB TO FILL BUFFER WITH 33 WORST CASE PATTERN.
1993  005612  004567  177340    FW336:  JSR     %5,BMOVE        ;6 CHARACTER PATTERN TO BUFFER
1994  005616  016632                    A33WP6
1995  005620  017623                    BLOCK1
1996  005622  000006                    6
1997  005624  004567  177326            JSR     %5,BMOVE        ;FILL BUFFER WITH PATTERN.
1998  005630  017623                    BLOCK1
1999  005632  017631                    BLOCK1+6
2000  005634  000102                    66.
2001  005636  004567  177314            JSR     %5,BMOVE
2002  005642  017623                    BLOCK1
2003  005644  017735                    BLOCK2
2004  005646  000110                    72.
2005  005650  000207                    RTS     %7              ;EXIT
2006
2007                                    ;SUB TO FILL BUFFER WITH 35 WORST CASE PATTERN.
2008  005652  004567  177300    FW356:  JSR     %5,BMOVE        ;6 CHARACTER PATTERN TO BUFFER
2009  005656  016640                    A35WP6
2010  005660  017623                    BLOCK1
2011  005662  000006                    6
2012  005664  004567  177266            JSR     %5,BMOVE        ;FILL BUFFER WITH PATTERN.
2013  005670  017623                    BLOCK1
2014  005672  017631                    BLOCK1+6
2015  005674  000102                    66.
2016  005676  004567  177254            JSR     %5,BMOVE
2017  005702  017623                    BLOCK1
2018  005704  017735                    BLOCK2
2019  005706  000110                    72.
2020  005710  000207                    RTS     %7              ;EXIT.
```

```
                                        ;PRGO - INPUT-OUTPUT LOGIC TESTS
005712  012767  005724  173222  PRGO:   MOV     #AT0,KSTART     ;ADDRESS OF 1ST ROUTINE TO KSTART.
005720  000167  174404          JMP     SRSET           ;GO GET STARTED.
                                        ;TEST ABILITY TO REFERENCE THE KEYBOARD/READER STATUS WORD (TKS)
005724  000000          AT0:    0                       ;TEST #.
005726  005756                  AT1                     ;NEXT TEST.
005730  001750                  1000.                   ;I COUNT.
005732  005742                  AT0A                    ;SCOPE ENTRY.
005734  012767  005752  172042        MOV     #AT0E,MACHER    ;SET UP MACHINE ERROR TRAP.
005742  005777  173150  AT0A:   TST     @TKS            ;REFERENCE CODER STATUS WORD.
005746  104012          AT0B:   CHAIN                   ;CHAIN
005750  000774                  BR      AT0A            ;REPEAT TEST.
005752  104003          AT0E:   ERROR                   ;ERROR. TRAPPED WHEN REFERENCING READER.
005754  000774                  BR      AT0B            ;STATUS WORD (TKS).
                                        ;TEST ABILITY TO REFERENCE THE KEYBOARD/READER BUFFER (TKB).
005756  000001          AT1:    1                       ;TEST #.
005760  006010                  AT2                     ;NEXT TEST.
005762  001750                  1000.                   ;I COUNT.
005764  005774                  AT1A                    ;SCOPE ENTRY.
005766  012767  006004  172010        MOV     #AT1E,MACHER    ;SET UP MACHINE ERROR TRAP
005774  005777  173120  AT1A:   TST     @TKB            ;REFERENCE READER BUFFER.
006000  104012          AT1B:   CHAIN                   ;CHAIN
006002  000774                  BR      AT1A            ;REPEAT TEST.
006004  104003          AT1E:   ERROR                   ;ERROR. TRAPPED WHEN REFERENCING
006006  000774                  BR      AT1B            ;READER BUFFER. (TKB).
                                        ;TEST ABILITY TO REFERENCE PUNCH/PRINTER STATUS WORD (TPS).
006010  000002          AT2:    2                       ;TEST #.
006012  006042                  AT3                     ;NEXT TEST
006014  001750                  1000.                   ;I COUNT.
006016  006026                  AT2A                    ;SCOPE ENTRY.
006020  012767  006036  171756        MOV     #AT2E,MACHER    ;SETUP MACHINE ERROR TRAP.
006026  005777  173070  AT2A:   TST     @TPS            ;REFERENCE PUNCH/PRINTER STATUS WORD.
006032  104012          AT2B:   CHAIN                   ;CHAIN
006034  000774                  BR      AT2A            ;REPEAT TEST.
006036  104003          AT2E:   ERROR                   ;ERROR. TRAPPED WHEN REFERENCING
006040  000774                  BR      AT2B            ;PUNCH/PRINTER STATUS WORD (TPS).
                                        ;TEST ABILITY TO REFERENCE PUNCH/PRINTER BUFFER (TPB).
006042  000003          AT3:    3
006044  006074                  AT4
006046  001750                  1000.
006050  006060                  AT3A
006052  012767  006070  171724        MOV     #AT3E,MACHER    ;SETUP MACHINE ERROR TRAP.
006060  005777  173040  AT3A:   TST     @TPB            ;REFERENCE PUNCH/PRINTER BUFFER.
006064  104012          AT3B:   CHAIN                   ;CHAIN
006066  000774                  BR      AT3A            ;REPEAT TEST.
006070  104003          AT3E:   ERROR                   ;ERROR. TRAPPED WHEN REFERENCING
006072  000774                  BR      AT3B            ;PUNCH/PRINTER BUFFER. (TPS).
```

```
2069
2070                                        ;TEST ABILITY TO SET AND CLEAR READER/KYBD ID BIT
2071  006074  000004                AT4:    4                       ;TEST #
2072  006076  006160                        AT5                     ;NEXT TEST
2073  006100  001750                        1000.                   ;I COUNT
2074  006102  006112                        AT4A                    ;SCOPE ENTRY
2075  006104  012767  000340  171664        MOV     #PRTY7,PSW      ;SET PRIORITY 7.
2076  006112  052777  000100  172776  AT4A: BIS     #BIT6,@TKS      ;SET ID BIT IN TKS.
2077  006120  032777  000100  172770        BIT     #BIT6,@TKS      ;CHECK ID BIT IN TKS
2078  006126  001002                        BNE     AT4B            ;BRANCH IF ID BIT IS SET.
2079  006130  104003                AT4E1:  ERROR                   ;ERROR 1 ID BIT NOT SET.
2080  006132  000410                        BR      AT4C
2081  006134  042777  000100  172754  AT4B: BIC     #BIT6,@TKS      ;CLEAR ID BIT IN TKS
2082  006142  032777  000100  172746        BIT     #BIT6,@TKS      ;CHECK ID BIT IN TKS.
2083  006150  001401                        BEQ     AT4C            ;BRANCH IF ID BIT IS CLEARED.
2084  006152  104003                AT4E2:  ERROR                   ;ERROR. ID BIT FAILED TO CLEAR.
2085  006154  104012                AT4C:   CHAIN                   ;CHAIN
2086  006156  000755                        BR      AT4A            ;REPEAT TEST.
2087                                        ;TEST ABILITY TO CLEAR ID BIT WITH RESET INSTRUCTION.
2088  006160  000005                AT5:    5                       ;TEST #
2089  006162  007050                        AT24                    ;NEXT TEST
2090  006164  000144                        100.                    ;I COUNT
2091  006166  006176                        AT5A                    ;SCOPE ENTRY.
2092  006170  012767  000340  171600        MOV     #PRTY7,PSW      ;SET PRIORITY 7.
2093  006176  052777  000100  172712  AT5A: BIS     #BIT6,@TKS      ;SET ID BIT IN TKS
2094  006204  104011                        SRESET                  ;RESET
2095  006206  032777  000100  172702        BIT     #BIT6,@TKS      ;TEST ID BIT.
2096  006214  001401                        BEQ     AT5B            ;BRANCH IF ID BIT IS CLEAR.
2097  006216  104003                AT5E:   ERROR                   ;ERROR. RESET FAILED TO CLEAR ID BIT.
2098  006220  104012                AT5B:   CHAIN                   ;CHAIN
2099  006222  000765                        BR      AT5A            ;REPEAT TEST.
2100                                        ;TEST THAT READER DONE BIT SETS BY 200 MSECS AFTER READER ENABLE
2101  006224  000011                AT11:   11                      ;TEST #
2102  006226  006250                        AT12                    ;NEXT TEST
2103  006230  000144                        100.                    ;ICOUNT
2104  006232  006234                        AT11A                   ;SCOPE ENTRY
2105  006234  104400                AT11A:  DELAY
2106  006236  000226                        150.                    ;WAIT 150 MSECS
2107  006240  004767  175452              JSR     %7,AREAD        ;ENABLE READER
2108  006244  104012                AT11B:  CHAIN                   ;CHAIN
2109  006246  000772                        BR      AT11A           ;REPEAT TEST
```

```
2110                                    ;TEST THAT DONE BIT READS RELIABLY.
2111  006250  000012                AT12:   12                      ;TEST #
2112  006252  006322                        AT13                    ;NEXT TEST
2113  006254  000012                        10.                     ;I COUNT
2114  006256  006260                        AT12A                   ;SCOPE ENTRY.
2115  006260  104400                AT12A:  DELAY                   ;WAIT 150 MSECS.
2116  006262  000226                        150.
2117  006264  004767  175426        AT12B:  JSR     %7,AREAD        ;ENABLE READER. COME BACK WHEN DONE SET.
2118  006270  012767  001750  172772        MOV     #1000.,CTRA     ;1000 TO CTRA
2119  006276  105777  172614        AT12C:  TSTB    @TKS            ;TEST FOR DONE
2120  006302  100402                        BMI     AT12D           ;BRANCH IF DONE FOUND SET.
2121  006304  104003                AT12E:  ERROR                   ;ERROR. DONE BIT NOT FOUND SET.
2122  006306  000403                        BR      AT12F
2123  006310  005367  172754        AT12D:  DEC     CTRA            ;CHECKED 1000 TIMES?
2124  006314  001370                        BNE     AT12C           ;BRANCH IF NOT 1000 YET.
2125  006316  104012                AT12F:  CHAIN                   ;CHAIN
2126  006320  000761                        BR      AT12B           ;REPEAT TEST.
2127                                    ;TEST THAT RESET CLEARS DONE BIT
2128  006322  000013                AT13:   13                      ;TEST #
2129  006324  006360                        AT14                    ;NEXT TEST
2130  006326  000144                        100.                    ;I COUNT
2131  006330  006332                        AT13A                   ;SCOPE ENTRY
2132  006332  104400                AT13A:  DELAY                   ;WAIT 150 MSECS.
2133  006334  000226                        150.
2134  006336  004767  175354        AT13B:  JSR     %7,AREAD        ;ENABLE READER. COME BACK WHEN DONE SET.
2135  006342  104011                        SRESET                  ;ISSUE RESET.
2136  006344  105777  172546                TSTB    @TKS            ;TEST FOR DONE BIT
2137  006350  100001                        BPL     AT13C           ;BRANCH IF DONE BIT RESET.
2138  006352  104003                AT13E:  ERROR                   ;ERROR. RESET FAILED TO CLEAR DONE.
2139  006354  104012                AT13C:  CHAIN                   ;CHAIN
2140  006356  000767                        BR      AT13B           ;REPEAT TEST.
2141                                    ;TEST THAT REFERENCING READER DATA BUFFER CLEARS DONE
2142  006360  000014                AT14:   14                      ;TEST #
2143  006362  006420                        AT17                    ;NEXT TEST
2144  006364  000144                        100.                    ;I COUNT
2145  006366  006370                        AT14A                   ;SCOPE ENTRY
2146  006370  104400                AT14A:  DELAY                   ;WAIT 150 MSECS.
2147  006372  000226                        150.
2148  006374  004767  175316        AT14B:  JSR     %7,AREAD        ;ENABLE READER. RETURN WHEN DONE SET.
2149  006400  105777  172514                TSTB    @TKB            ;REFERENCE READ BUFFER.
2150  006404  105777  172506                TSTB    @TKS            ;TEST FOR DONE BIT
2151  006410  100001                        BPL     AT14C           ;BRANCH IF DONE NOT SET.
2152  006412  104003                AT14E:  ERROR                   ;ERROR. REFERENCE TO BUFFER DID NOT RESET DONE.
2153  006414  104012                AT14C:  CHAIN                   ;CHAIN
2154  006416  000766                        BR      AT14B           ;REPEAT TEST
2155                                    ;TEST THAT READ BUFFER CAN BE READ RELIABLY.
2156  006420  000017                AT17:   17                      ;TEST#
2157  006422  006514                        AT20                    ;NEXT TEST
2158  006424  000144                        100.                    ;I COUNT
2159  006426  006430                        AT17A                   ;SCOPE ENTRY
2160  006430  104400                AT17A:  DELAY                   ;WAIT 150 MSECS
2161  006432  000226                        150.
2162  006434  004767  175256        AT17B:  JSR     %7,AREAD        ;ENABLE READER. RETURN WHEN DONE SET.
2163  006440  117767  172454  172611        MOVB    @TKB,CRBUF+1    ;BUFFER CONTENTS TO CRBUF+1
2164  006446  012767  000144  172614        MOV     #100.,CTRA      ;100 TO CTRA
2165  006454  117767  172440  172574  AT17C:  MOVB    @TKB,CRBUF      ;BUFFER CONTENTS TO CRBUF
```

```
 2166  006462  126767  172570  172567          CMPB    CRBUF,CRBUF+1   ;COMPARE CONTENTS OF CRBUF AND CRBUF+1
 2167  006470  001404                          BEQ     AT17D           ;BRANCH IF SAME.
 2168  006472  016700  172560                  MOV     CRBUF,%0        ;NOT SAME, ERROR, HALT WITH 1ST READ CHAR
 2169  006476  000000                  AT17E:  HALT                    ;IN DATA BYTES LEFT. SUBSEQUENT READ IN DATA BYTES RIGHT
 2170  006500  000403                          BR      AT17F
 2171  006502  005367  172562          AT17D:  DEC     CTRA            ;HERE IF SAME. CHECKED 100 TIMES?
 2172  006506  001362                          BNE     AT17C           ;BRANCH IF NOT.
 2173  006510  104012                  AT17F:  CHAIN                   ;CHAIN
 2174  006512  000750                          BR      AT17B           ;REPEAT TEST.
```

```
2175
2176                                    ;TEST THAT READER DONE BIT IS ABLE TO CAUSE INTERRUPT. IF THE INTERRUPT IS
2177                                    ;SERVICED, IT WILL HAVE OCURRED AT CORRECT VECTOR.
2178  006514  000020                    AT20:   20                      ;TEST#
2179  006516  006572                            AT21                    ;NEXT TEST
2180  006520  001750                            1000.                   ;I COUNT
2181  006522  006540                            AT20B                   ;SCOPE ENTRY
2182  006524  104006                            STRDRV                  ;SET UP READER VECTOR TO AT20C
2183  006526  006564                            AT20C
2184  006530  104400                    AT20A:  DELAY                   ;WAIT 150 MSECS.
2185  006532  000226                            150.
2186  006534  004767  175156                    JSR     %7,AREAD        ;ENABLE READER. RETURN WHEN DONE SET.
2187  006540  005077  172352            AT20B:  CLR     @TKS            ;DISABLE READER INTERRUPTS
2188  006544  005067  171226                    CLR     PSW             ;ENABLE READER. RETURN WHEN DONE SET.
2189  006550  052777  000100  172340            BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPT,
2190  006556  000240                            NOP
2191  006560  104003            AT20E:  ERROR                           ;ERROR. READER FAILED TO INTERRUPT.
2192  006562  000401                            BR      AT20D
2193  006564  022626            AT20C:  POPSP2                          ;HERE IF INTERRUPT OCCURS. POP STACK TWICE.
2194  006566  104012            AT20D:  CHAIN                           ;CHAIN
2195  006570  000763                            BR      AT20B           ;REPEAT TEST.
2196                                    ;TEST THAT DONE DOES NOT CAUSE INTERRUPT WITH PROCESSOR AT SAME
2197                                    ;PRIORITY LEVEL AS THE READERS INTERRUPT REQUEST LEVEL.
2198  006572  000021            AT21:   21                              ;TEST#
2199  006574  006656                            AT22                    ;NEXT TEST.
2200  006576  001750                            1000.                   ;I COUNT.
2201  006600  006616                            AT21B                   ;SCOPE ENTRY.
2202  006602  104006                            STRDRV                  ;SET READER VECTOR TO AT21E.
2203  006604  006650                            AT21E
2204  006606  104400                    AT21A:  DELAY                   ;WAIT 150 MSECS.
2205  006610  000226                            150.
2206  006612  004767  175100                    JSR     %7,AREAD        ;ENABLE READER. RETURN WHEN DONE SET.
2207  006616  005077  172274            AT21B:  CLR     @TKS            ;DISABLE READER INTERRUPTS.
2208  006622  016767  172302  171146            MOV     TKLVL,PSW       ;SET PROCESSOR TO SAME PRIORITY AS READER'S.
2209  006630  052777  000100  172260            BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPTS.
2210  006636  000240                            NOP                     ;NO OP.
2211  006640  005077  172252            AT21C:  CLR     @TKS            ;OK IF NO INTERRUPT OCCURS.
2212  006644  104012                            CHAIN                   ;CHAIN
2213  006646  000763                            BR      AT21B           ;REPEAT TEST.
2214  006650  022626            AT21E:  POPSP2                          ;ERROR. READER ERRONEOUSLY INTERRUPTED
2215                                                                    ;WITH PROCESSOR AT SAME PRIORITY
2216  006652  104003                            ERROR                   ;LEVEL AS THE READER, OR THE READER
2217  006654  000771                            BR      AT21C           ;IS AT HIGHER PRIORITY THAN SPECIFIED
2218                                                                    ;AT TKLVL.
2219
2220
2221                                    ;TEST THAT DONE CAUSES INTERRUPT WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
2222                                    ;THAN THE READER'S INTERRUPT PRIORITY LEVEL.
2223  006656  000022            AT22:   22                              ;TEST#
2224  006660  006750                            AT23                    ;NEXT TEST
2225  006662  001750                            1000.                   ;I COUNT
2226  006664  006736                            AT22B                   ;SCOPE ENTRY
2227  006666  104006                            STRDRV                  ;SET READER INTERRUPT SERVICE TO
2228  006670  006736                            AT22B                   ;AT22B.
2229  006672  104400                            DELAY                   ;WAIT 150 MSECS.
2230  006674  000226                            150.
```

```
2231  006676  004767  175014            JSR     %7,AREAD        ;ENABLE READER. RETURN WHEN DONE SET.
2232  006702  005077  172210  AT22A:    CLR     @TKS            ;DISABLE READER INTERRUPTS
2233  006706  016767  172216  171062    MOV     TKLVL,PSW       ;SET PROCESSOR PRIORITY ONE LEVEL LOWER
2234  006714  162767  000040  171054    SUB     #40,PSW         ;THAN READER.(SPECIFIED AT TKLVL).
2235  006722  052777  000100  172166    BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPTS.
2236  006730  000240                    NOP
2237  006732  104003            AT22E:  ERROR                   ;FAILED TO INTERRUPT WITH PC AT PRIORITY ONE LEVEL LOWER
2238  006734  000401                    BR      AT22C           ;THAN READER. THEREFORE, READER PRIORITY MUST BE LOWER
2239  006736  022626            AT22B:  POPSP2                  ;HERE IF INTERRUPT OCCURS. OK. POP STACK TWICE
2240  006740  005077  172152    AT22C:  CLR     @TKS            ;DISABLE READER INTERRUPTS
2241  006744  104012                    CHAIN                   ;CHAIN
2242  006746  000755                    BR      AT22A           ;REPEAT TEST
2243                            ;TEST THAT DONE DOES NOT REINTERRUPT AFTER RTI WHEN DONE IS NOT CLEARED.
2244  006750  000023            AT23:   23                      ;TEST#
2245  006752  010220                    AT43                    ;NEXT TEST
2246  006754  001750                    1000.                   ;I COUNT
2247  006756  006770                    AT23A                   ;SCOPE ENTRY.
2248  006760  104400                    DELAY                   ;WAIT 150 MSECS
2249  006762  000226                    150.
2250  006764  004767  174726            JSR     %7,AREAD        ;ENABLE READER. RETURN WHEN DONE SET.
2251  006770  104006            AT23A:  STRDRV                  ;SET READER INTERRUPT SERVICE
2252  006772  007022                    AT23C                   ;TO AT23C.
2253  006774  005077  172116            CLR     @TKS            ;DISABLE READER INTERRUPTS.
2254  007000  052777  000100  172110    BIS     #BIT6,@TKS      ;ENABLE READER INTERRUPTS.
2255  007006  000240                    NOP
2256  007010  104003            AT23E1: ERROR                   ;ERROR1 FAILED TO INTERRUPT
2257  007012  005077  172100    AT23B:  CLR     @TKS            ;DISABLE READER INTERRUPTS.
2258  007016  104012                    CHAIN                   ;CHAIN
2259  007020  000763                    BR      AT23A           ;REPEAT TEST
2260  007022  012777  007042  172076  AT23C:  MOV     #AT23E2,@TKVTR  ;CHANGE INTERRUPT VECTOR TO AT23E2
2261  007030  012716  007036            MOV     #AT23D,@%6
2262  007034  000002                    RTI                     ;RETURN FROM INTERRUPT
2263  007036  000240            AT23D:  NOP
2264  007040  000764                    BR      AT23B           ;OK IF NO REINTERRUPT OCCURS.
2265  007042  022626            AT23E2: POPSP2                  ;ERROR2. DONE REINTERRUPTED AFTER
2266  007044  104003                    ERROR                   ;RTI WITH DONE BIT LEFT ON.
2267  007046  000761                    BR      AT23B
2268                            ;TEST ABILITY TO SET AND CLEAR PUNCH ID BIT
2269  007050  000024            AT24:   24                      ;TEST#
2270  007052  007134                    AT25                    ;NEXT TEST.
2271  007054  001750                    1000.                   ;I COUNT
2272  007056  007066                    AT24A                   ;SCOPE ENTRY.
2273  007060  012767  000340  170710    MOV     #PRTY7,PSW      ;SET PRIORITY 7.
2274  007066  052777  000100  172026  AT24A:  BIS     #BIT6,@TPS      ;SET PUNCH ID BIT.
2275  007074  032777  000100  172020            BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT.
2276  007102  001002                    BNE     AT24B           ;BRANCH IF PUNCH ID BIT IS SET.
2277  007104  104003            AT24E1: ERROR                   ;ERROR1. PUNCH ID BIT DID NOT SET.
2278  007106  000410                    BR      AT24C
2279  007110  042777  000100  172004  AT24B:  BIC     #BIT6,@TPS      ;CLEAR PUNCH ID BIT.
2280  007116  032777  000100  171776            BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT.
2281  007124  001401                    BEQ     AT24C           ;BRANCH IF PUNCH ID BIT IS CLEAR
2282  007126  104003            AT24E2: ERROR                   ;ERROR2. PUNCH ID BIT FAILED TO CLEAR.
2283  007130  104012            AT24C:  CHAIN                   ;CHAIN
2284  007132  000776                    BR      AT24C           ;REPEAT TEST
2285                            ;TEST ABILITY TO CLEAR PUNCH ID BIT WITH RESET INSTRUCTION
2286  007134  000025            AT25:   25                      ;TEST#
```

```
2287  007136  007200                          AT26                    ;NEXT TEST.
2288  007140  000144                          100.                    ;I COUNT.
2289  007142  007152                          AT25A                   ;SCOPE ENTRY.
2290  007144  012767  000340  170624          MOV     #PRTY7,PSW      ;SET PRIORITY 7.
2291  007152  052777  000100  171742  AT25A:  BIS     #BIT6,@TPS      ;SET PUNCH ID BIT.
2292  007160  104011                          SRESET                  ;RESET
2293  007162  032777  000100  171732          BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT.
2294  007170  001401                          BEQ     AT25B           ;BRANCH IF PUNCH ID BIT IS CLEAR.
2295  007172  104003                  AT25E:  ERROR                   ;ERROR. RESET FAILED TO CLEAR PUNCH ID BIT.
2296  007174  104012                  AT25B:  CHAIN                   ;CHAIN
2297  007176  000765                          BR      AT25A           ;REPEAT TEST.
2298                                  ;TEST ABILITY TO SET AND CLEAR THE PUNCH MAINTENANCE BIT
2299  007200  000026                  AT26:   26                      ;TEST#
2300  007202  007256                          AT27                    ;NEXT TEST
2301  007204  001750                          1000.                   ;I COUNT
2302  007206  007210                          AT26A                   ;SCOPE ENTRY
2303  007210  052777  000004  171704  AT26A:  BIS     #BIT2,@TPS      ;SET MAINTANCE BIT.
2304  007216  032777  000004  171676          BIT     #BIT2,@TPS      ;CHECK MAINTENANCE BIT
2305  007224  001002                          BNE     AT26B           ;BRANCH IF MAINTENANCE BIT SET.
2306  007226  104003                  AT26E1: ERROR                   ;ERROR1. MAINTENANCE BIT FAILED TO SET.
2307  007230  000410                          BR      AT26C
2308  007232  042777  000004  171662  AT26B:  BIC     #BIT2,@TPS      ;CLEAR MAINTENANCE BIT.
2309  007240  032777  000004  171654          BIT     #BIT2,@TPS      ;CHECK MAINTENANCE BIT
2310  007246  001401                          BEQ     AT26C           ;BRANCH IF MAINTENANCE BIT IS CLEAR.
2311  007250  104003                  AT26E2: ERROR                   ;ERROR2. MAINTENANCE BIT FAILED TO CLEAR.
2312  007252  104012                  AT26C:  CHAIN                   ;CHAIN
2313  007254  000755                          BR      AT26A           ;REPEAT TEST
```

```
2314
2315                                        ;TEST THAT RESET INSTRUCTION CLEARS THE MAINTENANCE BIT.
2316  007256  000027                AT27:   27                      ;TEST#
2317  007260  007314                        AT30                    ;NEXT TEST
2318  007262  000144                        100.                    ;I COUNT
2319  007264  007266                        AT27A                   ;SCOPE ENTRY
2320  007266  052777  000004  171626 AT27A: BIS     #BIT2,@TPS      ;SET MAINTENANCE BIT.
2321  007274  104011                        SRESET                  ;ISSUE RESET
2322  007276  032777  000004  171616        BIT     #BIT2,@TPS      ;CHECK MAINTENANCE BIT
2323  007304  001401                        BEQ     AT27B           ;BRANCH IF MAINTENANCE BIT CLEAR.
2324  007306  104003                AT27E:  ERROR                   ;ERROR. RESET FAILED TO CLEAR
2325  007310  104012                AT27B:  CHAIN                   ;THE MAINTENANCE BIT. CHAIN.
2326  007312  000765                        BR      AT27A           ;REPEAT TEST.
2327                                        ;TEST THAT RESET SETS THE PUNCH READY BIT, AND THAT READY CAN BE READ RELIABLY.
2328  007314  000030                AT30:   30                      ;TEST#
2329  007316  007340                        AT31                    ;NEXT TEST
2330  007320  001750                        1000.                   ;I COUNT
2331  007322  007324                        AT30A                   ;SCOPE ENTRY
2332  007324  105777  171572        AT30A:  TSTB    @TPS            ;CHECK PUNCH READY.
2333  007330  100401                        BMI     AT30B           ;BRANCH IF PUNCH READY IS SET.
2334  007332  104003                AT30E:  ERROR                   ;ERROR. RESET FAILED TO SET READY, OR FAILED TO READ  IT
2335  007334  104012                AT30B:  CHAIN                   ;CHAIN
2336  007336  000772                        BR      AT30A           ;REPEAT TEST.
2337                                        ;TEST THAT PUNCH READY RESETS BY LOADING PUNCH BUFFER.
2338  007340  000031                AT31:   31                      ;TEST#
2339  007342  007376                        AT32                    ;NEXT TEST
2340  007344  000024                        20.                     ;I COUNT
2341  007346  007350                        AT31A                   ;SCOPE ENTRY
2342  007350  104400                AT31A:  DELAY                   ;WAIT 150 MSECS
2343  007352  000226                        150.
2344  007354  104011                        SRESET                  ;RESET
2345  007356  005077  171542               CLR     @TPB            ;LOAD PUNCH BUFFER
2346  007362  105777  171534               TSTB    @TPS            ;CHECK PUNCH READY BIT.
2347  007366  100001                        BPL     AT31B           ;BRANCH IF PUNCH READY IS CLEAR.
2348  007370  104003                AT31E:  ERROR                   ;ERROR. BUFFER LOAD FAILED TO CLEAR READY.
2349  007372  104012                AT31B:  CHAIN                   ;CHAIN
2350  007374  000765                        BR      AT31A           ;REPEAT TEST.
2351                                        ;TEST THAT BYTE LOAD OF PUNCH BUFFER +1 DOES NOT RESET READY.
2352  007376  000032                AT32:   32                      ;TEST#
2353  007400  007440                        AT33                    ;NEXT TEST
2354  007402  000024                        20.                     ;I COUNT
2355  007404  007406                        AT32A                   ;SCOPE ENTRY
2356  007406  104400                AT32A:  DELAY                   ;WAIT 150 MSECS
2357  007410  000226                        150.
2358  007412  104011                        SRESET                  ;RESET
2359  007414  016700  171504               MOV     TPB,%0
2360  007420  005200                        INC     %0
2361  007422  105010                        CLRB    @%0             ;BYTE LOAD PUNCH BUFFER+1
2362  007424  105777  171472               TSTB    @TPS            ;CHECK PUNCH READY BIT
2363  007430  100401                        BMI     AT32B           ;BRANCH IF PUNCH READY STILL SET.
2364  007432  104003                AT32E:  ERROR                   ;ERROR. BYTE LOAD OF PUNCH BUFFER+1
2365  007434  104012                AT32B:  CHAIN                   ;CLEARED READY. CHAIN
2366  007436  000763                        BR      AT32A           ;REPEAT TEST.
```

```
2367
2368                                    ;TEST THAT PUNCH BECOMES READY BY 200 MSECS AFTER BUFFER LOAD.
2369  007440  000033                    AT33:   33                      ;TEST #
2370  007442  007500                            AT34                    ;NEXT TEST
2371  007444  000024                            20.                     ;I COUNT
2372  007446  007450                            AT33A                   ;SCOPE ENTRY.
2373  007450  104400                    AT33A:  DELAY                   ;WAIT 150 MSECS.
2374  007452  000226                            150.
2375  007454  005077  171444                    CLR     @TPB            ;LOAD PUNCH BUFFER.
2376  007460  104400                            DELAY                   ;WAIT 200 MSECS.
2377  007462  000310                            200.
2378  007464  105777  171432                    TSTB    @TPS            ;CHECK PUNCH READY BIT.
2379  007470  100401                            BMI     AT33B           ;BRANCH IF PUNCH READY IS SET.
2380  007472  104003                    AT33E:  ERROR                   ;ERROR. READY NOT SET 200 MSECS AFTER BUFFER LOAD.
2381  007474  104012                    AT33B:  CHAIN                   ;CHAIN
2382  007476  000764                            BR      AT33A           ;REPEAT TEST.
2383                                    ;TEST THAT PUNCH READY BIT CAN CAUSE AN INTERRUPT. IF THE INTERRUPT
2384                                    ;IS SERVICED, IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
2385  007500  000034                    AT34:   34                      ;TEST #
2386  007502  007546                            AT35                    ;NEXT TEST
2387  007504  001750                            1000.                   ;I COUNT
2388  007506  007514                            AT34A                   ;SCOPE ENTRY
2389  007510  104007                            STPCHV                  ;SET PUNCH INTERRUPT SERVICE
2390  007512  007542                            AT34C                   ;TO AT34C
2391  007514  005077  171402            AT34A:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
2392  007520  005067  170252                    CLR     PSW             ;SET PRIORITY 0.
2393  007524  052777  000100  171370            BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS.
2394  007532  000240                            NOP
2395  007534  104003                    AT34E:  ERROR                   ;PUNCH READY FAILED TO CAUSE
2396  007536  104012                    AT34B:  CHAIN                   ;INTERRUPT. CHAIN
2397  007540  000765                            BR      AT34A           ;REPEAT TEST.
2398  007542  022626                    AT34C:  POPSP2                  ;HERE IF INTERRUPT OCCURS. POP THE
2399  007544  000774                            BR      AT34B           ;STOCK TWICE.
2400                                    ;TEST THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WITH PROCESSOR
2401                                    ;AT SAME PRIORITY LEVEL AS THE PUNCH INTERRUPT REQUEST LEVEL.
2402  007546  000035                    AT35:   35                      ;TEST #
2403  007550  007622                            AT36                    ;NEXT TEST
2404  007552  001750                            1000.                   ;I COUNT
2405  007554  007562                            AT35A                   ;SCOPE ENTRY
2406  007556  104007                            STPCHV                  ;SET PUNCH INTERRUPT SERVICE
2407  007560  007614                            AT35E                   ;TO AT35E.
2408  007562  016767  171346  170206    AT35A:  MOV     TPLVL,PSW       ;SET PROCESSOR TO SAME PRIORITY AS PUNCH.
2409  007570  005077  171326                    CLR     @TPS            ;DISABLE PUNCH INTERRUPTS.
2410  007574  052777  000100  171320            BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS.
2411  007602  000240                            NOP
2412  007604  005077  171312            AT35B:  CLR     @TPS            ;OK IF NO INTERRUPT OCCURS.
2413  007610  104012                            CHAIN                   ;CHAIN
2414  007612  000763                            BR      AT35A           ;REPEAT TEST.
2415  007614  022626                    AT35E:  POPSP2                  ;ERROR. PUNCH INTERRUPTED WITH PROCESSOR
2416  007616  104003                            ERROR                   ;SET TO SAVE PRIORITY AS THE PUNCH.
2417  007620  000771                            BR      AT35B
```

```
2418
2419                                    ;TEST THAT THE PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
2420                                    ;THAN THE PUNCH PRIORITY.
2421  007622  000036                    AT36:   36                      ;TEST #
2422  007624  007704                            AT37                    ;NEXT TEST
2423  007626  001750                            1000.                   ;I COUNT
2424  007630  007636                            AT36A                   ;SCOPE ENTRY
2425  007632  104007                            STPCHV                  ;SET PUNCH INTERRUPT SERVICE
2426  007634  007672                            AT36B                   ;TO AT36B.
2427  007636  005077  171260           AT36A:   CLR     @TPS           ;DISABLE PUNCH INTERRUPTS
2428  007642  016767  171266  170126           MOV     TPLVL,PSW       ;SET PROCESSOR PRIORITY ONE LEVEL
2429  007650  162767  000040  170120           SUB     #40,PSW         ;LOWER THAN PUNCH PRIORITY
2430  007656  052777  000100  171236           BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
2431  007664  000240                            NOP
2432  007666  104003                            ERROR                   ;ERROR. PUNCH FAILED TO INTERRUPT.
2433  007670  000401                            BR      AT36C
2434  007672  022626                    AT36B:  POPSP2                  ;HERE IF INTERRUPT OCCURS. POP
2435  007674  005077  171222           AT36C:  CLR     @TPS            ;THE STOCK TWICE. DISABLE PUNCH INTERRUPT
2436  007700  104012                            CHAIN                   ;CHAIN
2437  007702  000755                            BR      AT36A           ;REPEAT TEST.
2438                                    ;TEST THAT PUNCH READY DOES NOT REINTERRUPT AFTER RTI WHEN READY
2439                                    ;BIT HAS NOT BEEN RESET.
2440  007704  000037                    AT37:   37                      ;TEST #
2441  007706  010000                            AT40                    ;NEXT TEST
2442  007710  001750                            1000.                   ;I COUNT
2443  007712  007714                            AT37A                   ;SCOPE ENTRY
2444  007714  104007                    AT37A:  STPCHV                  ;SET PUNCH INTERRUPT SERVICE TO
2445  007716  007752                            AT37C                   ;AT37C
2446  007720  005077  171176                    CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
2447  007724  005067  170046                    CLR     PSW             ;SET PROCESSOR PRIORITY TO 0
2448  007730  052777  000100  171164           BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
2449  007736  000240                            NOP
2450  007740  104003                    AT37E1: ERROR                   ;ERROR 1. PUNCH FAILED TO INTERRUPT.
2451  007742  005077  171154           AT37B:  CLR     @TPS            ;DISABLE PUNCH INTERRUPT.
2452  007746  104012                            CHAIN                   ;CHAIN
2453  007750  000761                            BR      AT37A           ;REPEAT TEST.
2454  007752  012777  007772  171152   AT37C:  MOV     #AT37E2,@TPVTR  ;HERE IF INTERRUPT OCCURS. CHANGE
2455  007760  012716  007766                    MOV     #AT37D,@%6      ;PUNCH VECTOR TO AT37E2 AND EXIT
2456  007764  000002                            RTI                     ;INTERRUPT
2457  007766  000240                    AT37D:  NOP                     ;OK IF NO REINTERRUPT OCCURS
2458  007770  000764                            BR      AT37B
2459  007772  022626                    AT37E2: POPSP2                  ;ERROR 2.  PUNCH REINTERRUPTED AFTER
2460  007774  104003                            ERROR                   ;RTI WITH READY BIT LEFT ON
2461  007776  000761                            BR      AT37B
```

```
2462
2463                                    ;TEST THAT THE PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING
2464                                    ;PROCESSOR PRIORITY TO 0.
2465  010000  000040                    AT40:   40                      ;TEST #
2466  010002  010064                            AT41                    ;NEXT TEST
2467  010004  001750                            1000.                   ;I COUNT
2468  010006  010014                            AT40A                   ;SCOPE ENTRY
2469  010010  104007                            STPCHV                  ;SET PUNCH INTERRUPT
2470  010012  010052                            AT40B                   ;SERVICE TO AT40B
2471  010014  012767  000340  167754  AT40A:    MOV     #PRTY7,PSW      ;SET PROCESSOR PRIORITY TO 7.
2472  010022  005077  171074                    CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
2473  010026  052777  000100  171066            BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
2474  010034  005067  167736                    CLR     PSW             ;LOWER PROCESSOR PRIORITY TO 0.
2475  010040  012767  000340  167730            MOV     #PRTY7,PSW      ;RAISE PRIORITY TO 7.
2476  010046  104003                            ERROR                   ;ERROR. PUNCH FAILED TO INTERRUPT
2477  010050  000401                            BR      AT40C           ;IMMEDIATELY AFTER CP PRIORITY WAS SET TO 0.
2478  010052  022626                    AT40B:  POPSP2                  ;HERE IF INTERRUPT OCCURS
2479  010054  005077  171042            AT40C:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
2480  010060  104012                            CHAIN                   ;CHAIN
2481  010062  000754                            BR      AT40A           ;REPEAT TEST
2482                                    ;TEST FOR CORRECT OPERATION OF THE WAIT INSTRUCTION.  A WAIT INSTRUCTION
2483                                    ;IS PERFORMED WHILE WAITING FOR A PUNCH INTERRUPT.  WHEN THE INTERRUPT
2484                                    ;OCCURS, THE SERVICE ROUTINE CHANGES THE WAIT INSTRUCTION TO AN ERROR
2485                                    ;CALL AND THEN EXITS THE INTERRUPT WITH AN RTI.  EXITING THE INTERRUPT
2486                                    ;SHOULD RETURN CONTROL TO THE INSTRUCTION FOLLOWING THE WAIT INSTRUCTION.
2487                                    ;IF CONTROL IS INSTEAD RETURNED TO THE SAME LOCATION WHERE THE WAIT
2488                                    ;INSTRUCTION WAS LOCATED AN ERROR CALL WILL OCCUR, INDICATING A FAILURE
2489                                    ;OF THE WAIT INSTRUCTION.
2490  010064  000041                    AT41:   41                      ;TEST#
2491  010066  010152                            AT42                    ;NEXT TEST
2492  010070  000062                            50.                     ;I COUNT
2493  010072  010104                            AT41A                   ;SCOPE ENTRY
2494  010074  104400                            DELAY                   ;WAIT 150 MSECS
2495  010076  000226                            150.
2496  010100  104007                            STPCHV                  ;SET PUNCH INTERRUPT SERVICE
2497  010102  010142                            AT41C                   ;TO AT41C
2498  010104  012767  000001  000016  AT41A:    MOV     #WAIT,AT41B     ;MOVE WAIT INSTRUCTION TO AT41B
2499  010112  005077  171006                    CLR     @TPB            ;LOAD PUNCH BUFFER (ENABLES PUNCH)
2500  010116  052777  000100  170776            BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
2501  010124  005067  167646                    CLR     PSW             ;SET PRIORITY 0.
2502  010130  000000                    AT41B:  OPEN                    ;THIS LOCATION CAN BE EITHER
2503                                            ;A WAIT INSTRUCTION OR AN ERROR CALL.
2504                                            ;IF AN ERROR CALL IS EXECUTED, IT
2505                                            ;INDICATES A FAILURE OF THE WAIT INSTRUCTION.
2506  010132  005077  170764                    CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
2507  010136  104012                            CHAIN                   ;CHAIN
2508  010140  000761                            BR      AT41A           ;REPEAT TEST
2509  010142  012767  104003  177760  AT41C:    MOV     #ERROR,AT41B    ;MOVE ERROR CALL TO AT41B.
2510  010150  000002                            RTI                     ;EXIT INTERRUPT.
```

```
2511
2512                                    ;TEST THAT LOADING THE PUNCH BUFFER WITH THE MAINTENANCE BIT SET
2513                                    ;CAUSES THE READER DONE BIT TO SET AFTER APPROX. 200 MSECS
2514  010152  000042                   AT42:   42                      ;TEST #
2515  010154  006224                           AT11                    ;NEXT TEST
2516  010156  000062                           50.                     ;I COUNT
2517  010160  010166                           AT42A                   ;SCOPE ENTRY
2518  010162  104400                           DELAY
2519  010164  000226                           150.
2520  010166  052777  000004  170726   AT42A:  BIS     #BIT2,@TPS      ;SET MAINTENANCE BIT
2521  010174  005077  170724                   CLR     @TPB            ;LOAD PUNCH BUFFER
2522  010200  104400                           DELAY                   ;WAIT 200 MSECS
2523  010202  000310                           200.
2524  010204  105777  170706                   TSTB    @TKS            ;TEST READER DONE BIT
2525  010210  100401                           BMI     AT42B           ;BRANCH IF READER DONE BIT SET.
2526  010212  104003                   AT42E:  ERROR                   ;ERROR. 200 MSECS AFTER PUNCH
2527                                                                   ;BUFFER LOAD WITH MAINTENANCE BIT
2528                                                                   ;SET THE READER DONE BIT WAS NOT SET
2529  010214  104012                   AT42B:  CHAIN                   ;CHAIN
2530  010216  000763                           BR      AT42A           ;REPEAT TEST
2531                                    ;TESTS THAT DATA CAN BE RELIABLY
2532                                    ;TRANSFERRED WITH THE MAINTENANCE BIT
2533                                    ;SET.  CAUSES PROGRAM 7 TO BE
2534                                    ;EXECUTED AS A SUBROUTINE IF THIS
2535                                    ;WAS A MONITOR LOAD OR IF THE
2536                                    ;MONITOR CELL IS NON-ZERO
2537  010220  000043                   AT43:   43                      ;TEST #
2538  010222  177777                           -1                      ;LAST TEST
2539  010224  001750                           1000.                   ;ICOUNT
2540  010226  012562                           LTA                     ;SCOPE ENTRY
2541  010230  005767  167606                   TST     MONITR          ;IS THIS AUTOMATIC TESTING
2542  010234  001402                           BEQ     AT43B           ;NO - EXIT
2543  010236  004767  002300           AT43A:  JSR     %7,PRG7         ;RUN PROGRAM 7
2544  010242  104012                   AT43B:  CHAIN                   ;CHAIN
2545  010244  005767  167572                   TST     MONITR
2546  010250  001774                           BEQ     AT43B
2547  010252  004767  002304                   JSR     %7,LTA          ;REPEAT TEST
2548  010256  000771                           BR      AT43B
```

```
2549
2550                                        ;PRG1-PRINTER TESTS
2551  010260  012767  010312  170654  PRG1: MOV     #CT0,KSTART     ;SET ADDRESS IF 1ST ROUTINE.
2552  010266  052767  000200  170662        BIS     #BIT7,PRGID     ;BYPASS SCOPE AND ICNT.
2553  010274  012767  177600  174314        MOV     #177600,STLMSK  ;SET STALL LIMIT
2554  010302  004767  175162                JSR     %7,STBF         ;SET UP BUFFER AREA.
2555  010306  000167  172016                JMP     SRSET           ;GO GET STARTED.
2556                                        ;CARRIAGE RETURN TEST.
2557  010312  000000                CT0:    0                       ;TEST#
2558  010314  010414                        CT1                     ;NEXT TEST ADDRESS.
2559  010316  104000                        TYPE                    ;TYPE TITLE.
2560  010320  017036                        CRTST
2561  010322  012767  000111  170724        MOV     #73.,RCNT
2562  010330  016767  170720  170732        MOV     RCNT,CTRA       ;RCNT TO CTRA
2563  010336  005367  170726          CT0A: DEC     CTRA            ;DECREMENT CTRA
2564  010342  001001                        BNE     CT0B            ;BRANCH IF NOT 0
2565  010344  104012                        CHAIN                   ;0. CHAIN
2566  010346  016767  170716  170716  CT0B: MOV     CTRA,CTRB       ;SPACE COUNT TO CTRB.
2567  010354  112700  000105          CT0C: MOVB    #105,%0         ;CHAR=E
2568  010360  004767  174630                JSR     %7,LSPCH        ;SPACE.
2569  010364  005367  170702                DEC     CTRB            ;DECREMENT CTRB.
2570  010370  001371                        BNE     CT0C            ;BRANCH IF NOT DONE SPACING.
2571  010372  112700  000015                MOVB    #15,%0
2572  010376  004767  174612                JSR     %7,LSPCH        ;CARRIAGE RETURN.
2573  010402  012700  000012                MOV     #12,%0          ;LINE FEED
2574  010406  004767  174602                JSR     %7,LSPCH        ;
2575  010412  000751                        BR      CT0A
2576                                        ;RIGHT MARGIN TEST
2577  010414  000001                CT1:    1                       ;TEST#
2578  010416  010460                        CT2                     ;NEXT TEST.
2579  010420  104000                        TYPE                    ;TYPE TITLE
2580  010422  017067                        RMTST
2581  010424  012767  000016  170636        MOV     #14.,CTRA       ;SET UP FOR 33/35
2582  010432  012767  017017  000014        MOV     #RM33B,RMB
2583  010440  104000                CT1A:   TYPE                    ;TYPE----I
2584  010442  017011                        RM33A
2585  010444  005367  170620                DEC     CTRA            ;DONE N TIMES.
2586  010450  001373                        BNE     CT1A            ;BRANCH IF NOT N TIMES
2587  010452  104000                        TYPE                    ;TYPE-I-.
2588  010454  000000                RMB:    OPEN
2589  010456  104012                        CHAIN                   ;CHAIN.
```

```
                                        ;CURSOR RIGHT TEST
010460  000002                  CT2:    2                       ;TEST#
010462  010632                          CT3                     ;NEXT TEST
010464  104000                          TYPE                    ;TYPE TITLE.
010466  017115                          SPTST
010470  012767  000044  170572          MOV     #36.,CTRA       ;33/35 COUNT TO CTRA.
010476  104000                  CT2A:   TYPE                    ;TYPE SPACE,\.
010500  017033                          SPTSTC
010502  005367  170562                  DEC     CTRA            ;DONE TIMES SET IN CTRA?
010506  001373                          BNE     CT2A            ;BRANCH IF NOT DONE
010510  012767  000044  170552          MOV     #36.,CTRA       ;SET UP CTRA COUNT FOR 33/35
010516  012767  000001  170546  CT2B:   MOV     #1,CTRB
010524  016767  170542  170542  CT2C:   MOV     CTRB,CTRC
010532  112700  000015                  MOVB    #15,%0          ;CARRIAGE RETURN.
010536  004767  174452                  JSR     %7,LSPCH
010542  004767  174446                  JSR     %7,LSPCH        ;DUMMY CYCLE.
010546  005000                  CT2D:   CLR     %0              ;NULL CHAR FOR FILLER
010550  004767  174440                  JSR     %7,LSPCH        ;TRANSMIT NULL CHAR
010554  004767  174434                  JSR     %7,LSPCH        ;TRANSMIT NULL CHAR
010560  004767  174430                  JSR     %7,LSPCH        ;TRANSMIT NULL CHAR
010564  112700  000030                  MOVB    #30,%0          ;CURSOR RIGHT
010570  004767  174420                  JSR     %7,LSPCH        ;SET IN CTRC.
010574  005367  170474                  DEC     CTRC            ;DONE SPACING.
010600  001362                          BNE     CT2D            ;BRANCH IF NOT DONE SPACING.
010602  112700  000057                  MOVB    #'/,%0          ;DONE. TYPE A "/".
010606  004767  174402                  JSR     %7,LSPCH
010612  005367  170452                  DEC     CTRA            ;DONE 36 TIMES?
010616  001001                          BNE     CT2E            ;BRANCH IF NOT DONE.
010620  104012                          CHAIN                   ;DONE. CHAIN.
010622  062767  000002  170442  CT2E:   ADD     #2,CTRB         ;MODIFY CTRB FOR NEXT TRY.
010630  000735                          BR      CT2C            ;GO DO IT AGAIN.
                                        ;LINE FEED TEST
010632  000003                  CT3:    3                       ;TEST #
010634  010712                          CT4                     ;NEXT TEST.
010636  104000                          TYPE                    ;TYPE TITLE
010640  017143                          LFTST
010642  052767  040000  170306          BIS     #BIT14,PRGID    ;ALLOW STALLS.
010650  012767  000110  170412          MOV     #72.,CTRA       ;SET 33/35 LINE FEED COUNT.
010656  112700  000134          CT3A:   MOVB    #'\,%0          ;TYPE "\"
010662  004767  174326                  JSR     %7,LSPCH
010666  112700  000012                  MOVB    #12,%0          ;LINE FEED.
010672  004767  174316                  JSR     %7,LSPCH
010676  005367  170366                  DEC     CTRA            ;DONE N TIMES?
010702  001001                          BNE     CT3B            ;BRANCH IF NOT DONE.
010704  104012                          CHAIN                   ;DONE. CHAIN
010706  104002                  CT3B:   STALL                   ;STALL
010710  000762                          BR      CT3A            ;REPEAT
```

```
                                         ;TAB TEST
  010712  000004                         CT4:    4                         ;TEST#
  010714  011130                                 CT5                       ;NEXT TEST.
  010716  012767  000011  000100                 MOV     #9.,TBCNT         ;SET TAB COUNT.
  010724  104014                                 CK35                      ;35?
  010726  104012                                 CHAIN                     ;NO.
  010730  004567  000044                         JSR     %5,TPBM           ;TYPE MARKERS
  010734  000007                                 7
  010736  104000                                 TYPE
  010740  016765                                 TBMRK+1
  010742  012767  000007  170320 CT4A:           MOV     #7,CTRA           ;LINE COUNT TO CTRA
  010750  005067  000052                         CLR     SPCNT             ;0 TO SPACE COUNT.
  010754  004767  000050         CT4B:           JSR     %7,TABP           ;GO SPACE-TAB.
  010760  005267  000042                         INC     SPCNT             ;INCREMENT SPACE COUNT.
  010764  005367  170300                         DEC     CTRA              ;DONE 7 LINES?
  010770  001371                                 BNE     CT4B              ;BRANCH IF NOT DONE.
  010772  004767  174402                         JSR     %7,TYPCLR         ;CLEAR DOUBLE BUFFER
  010776  104012                                 CHAIN                     ;DONE. CHAIN.
  011000  012567  170264         TPBM:           MOV     (5)+,CTRA         ;TYPE TEST TITLE.
  011004  104000                                 TYPE
  011006  016750                                 TBTST
  011010  104000                 TPBMA:          TYPE                      ;TYPE MARKERS
  011012  016776                                 TBMRK1
  011014  005367  170250                         DEC     CTRA
  011020  001373                                 BNE     TPBMA
  011022  000205                                 RTS     %5                ;EXIT.
  011024  000000                 TBCNT:          OPEN                      ;TAB COUNT
  011026  000000                 SPCNT:          OPEN                      ;SPACE COUNT
  011030  104000                 TABP:           TYPE                      ;CRLF.
  011032  017007                                 CRLF
  011034  016767  177764  170230                 MOV     TBCNT,CTRB        ;TAB COUNT TO CTRB
  011042  016767  177760  170224 TABPA:          MOV     SPCNT,CTRC        ;SPACE COUNT TO CTRC
  011050  001407                                 BEQ     TABPC             ;BRANCH IF SPACE COUNT IS 0.
  011052  112700  000040         TABPB:          MOVB    #40,%0            ;SPACE
  011056  004767  174132                         JSR     %7,LSPCH
  011062  005367  170206                         DEC     CTRC              ;DECREMENT SPACE COUNT
  011066  001371                                 BNE     TABPB             ;BRANCH IF NOT YET 0.
  011070  112700  000011         TABPC:          MOVB    #11,%0            ;TAB
  011074  004767  174114                         JSR     %7,LSPCH
  011100  004767  174110                         JSR     %7,LSPCH          ;DUMMY CYCLE
  011104  004767  174104                         JSR     %7,LSPCH          ;DUMMY CYCLE.
  011110  112700  000057                         MOVB    #'/,%0            ;TYPE "/"
  011114  004767  174074                         JSR     %7,LSPCH
  011120  005367  170146                         DEC     CTRB              ;DECREMENT TAB COUNT.
  011124  001346                                 BNE     TABPA             ;BRANCH IF NOT DONE TABBING.
  011126  000207                                 RTS     %7                ;DONE. EXIT.
```

```
 2683
 2684                                   ;TYPE LINE OF CHARACTERS ABC
 2685  011130  000005                   CT5:    5                       ;TEST #
 2686  011132  011146                           CT6                     ;NEXT TEST
 2687  011134  104000                           TYPE                    ;TYPE "CHARACTER TESTS"
 2688  011136  017166                           CHRTST
 2689  011140  104016                           TYPLN3                  ;TYPE LINE
 2690  011142  016646                           A
 2691  011144  104012                           CHAIN                   ;CHAIN
 2692                                   ;TYPE LINE OF CHARACTERS DEF
 2693  011146  000006                   CT6:    6                       ;TEST #
 2694  011150  011160                           CT7                     ;NEXT TEST
 2695  011152  104016                           TYPLN3                  ;TYPE LINE
 2696  011154  016651                           D
 2697  011156  104012                           CHAIN                   ;CHAIN
 2698                                   ;TYPE LINE OF CHARACTERS GHI
 2699  011160  000007                   CT7:    7                       ;TEST#
 2700  011162  011172                           CT10                    ;NEXT TEST.
 2701  011164  104016                           TYPLN3                  ;TYPE LINE
 2702  011166  016654                           G
 2703  011170  104012                           CHAIN                   ;CHAIN
 2704                                   ;TYPE LINE OF CHARACTERS OF JKL
 2705  011172  000010                   CT10:   10                      ;TEST #
 2706  011174  011204                           CT11                    ;NEXT TEST.
 2707  011176  104016                           TYPLN3                  ;TYPELINE
 2708  011200  016657                           J
 2709  011202  104012                           CHAIN                   ;CHAIN
 2710                                   ;TYPE LINE OF CHARACTERS MNO
 2711  011204  000011                   CT11:   11                      ;TEST #
 2712  011206  011216                           CT12                    ;NEXT TEST
 2713  011210  104016                           TYPLN3                  ;TYPE LINE
 2714  011212  016662                           M
 2715  011214  104012                           CHAIN                   ;CHAIN
 2716                                   ;TYPE LINE OF CHARACTERS PQR
 2717
 2718  011216  000012                   CT12:   12                      ;TEST #
 2719  011220  011230                           CT13                    ;NEXT TEST
 2720  011222  104016                           TYPLN3                  ;TYPE LINE
 2721  011224  016665                           P
 2722  011226  104012                           CHAIN                   ;CHAIN
 2723                                   ;TYPE LINE OF CHARACTERS STU
 2724  011230  000013                   CT13:   13                      ;TEST #
 2725  011232  011242                           CT14                    ;NEXT TEST
 2726  011234  104016                           TYPLN3
 2727  011236  016670                           S
 2728  011240  104012                           CHAIN
```

```
2729
2730                                    ;TYPE LINE OF CHARACTERS VWX
2731  011242  000014                    CT14:   14                      ;TEST #
2732  011244  011254                            CT15                    ;NEXT TEST
2733  011246  104016                            TYPLN3                  ;TYPE LINE
2734  011250  016673                            V
2735  011252  104012                            CHAIN                   ;CHAIN
2736                                    ;TYPE LINE OF CHARACTERS YZ0
2737  011254  000015                    CT15:   15                      ;TEST #
2738  011256  011266                            CT16                    ;NEXT TEST
2739  011260  104016                            TYPLN3                  ;TYPE LINE
2740  011262  016676                            Y
2741  011264  104012                            CHAIN                   ;CHAIN
2742                                    ;TYPE LINE OF CHARACTERS 123
2743  011266  000016                    CT16:   16                      ;TEST #
2744  011270  011300                            CT17                    ;NEXT TEST
2745  011272  104016                            TYPLN3                  ;TYPE LINE
2746  011274  016701                            ONE
2747  011276  104012                            CHAIN                   ;CHAIN
2748                                    ;TYPE LINE OF CHARACTERS 456
2749  011300  000017                    CT17:   17                      ;TEST #
2750  011302  011312                            CT20                    ;NEXT TEST
2751  011304  104016                            TYPLN3                  ;TYPE LINE
2752  011306  016704                            FOUR
2753  011310  104012                            CHAIN                   ;CHAIN
2754                                    ;TYPE LINE OF CHARACTERS 789
2755  011312  000020                    CT20:   20                      ;TEST #
2756  011314  011324                            CT21                    ;NEXT TEST
2757  011316  104016                            TYPLN3                  ;TYPE LINE
2758  011320  016707                            SEVEN
2759  011322  104012                            CHAIN                   ;CHAIN
2760                                    ;TYPE LINE OF CHARACTERS!"#
2761  011324  000021                    CT21:   21                      ;TEST #
2762  011326  011336                            CT22                    ;NEXT TEST
2763  011330  104016                            TYPLN3                  ;TYPE LINE
2764  011332  016712                            C41
2765  011334  104012                            CHAIN                   ;CHAIN
2766                                    ;TYPE LINE OF CHARACTERS $%&
2767  011336  000022                    CT22:   22                      ;TEST #
2768  011340  011350                            CT23                    ;NEXT TEST
2769  011342  104016                            TYPLN3                  ;TYPE LINE
2770  011344  016715                            C44
2771  011346  104012                            CHAIN                   ;CHAIN
2772                                    ;TYPE LINE OF CHARACTERS '()
2773  011350  000023                    CT23:   23                      ;TEST #
2774  011352  011362                            CT24                    ;NEXT TEST
2775  011354  104016                            TYPLN3                  ;TYPE LINE
2776  011356  016720                            C47
2777  011360  104012                            CHAIN                   ;CHAIN.
```

```
2778
2779                                    ;TYPE LINE OF CHARACTERS *+,
2780  011362  000024                   CT24:   24                      ;TEST #
2781  011364  011374                           CT25                    ;NEXT TEST
2782  011366  104016                           TYPLN3                  ;TYPE LINE
2783  011370  016723                           C52
2784  011372  104012                           CHAIN                   ;CHAIN
2785                                    ;TYPE LINE OF CHARACTERS -./
2786  011374  000025                   CT25:   25                      ;TEST #
2787  011376  011406                           CT26                    ;NEXT TEST
2788  011400  104016                           TYPLN3                  ;TYPE LINE
2789  011402  016726                           C55
2790  011404  104012                           CHAIN                   ;CHAIN
2791                                    ;TYPE LINE OF CHARACTERS :;<
2792  011406  000026                   CT26:   26                      ;TEST #
2793  011410  011420                           CT27                    ;NEXT TEST
2794  011412  104016                           TYPLN3                  ;TYPE LINE
2795  011414  016731                           C72
2796  011416  104012                           CHAIN                   ;CHAIN
2797                                    ;TYPE LINE OF CHARACTERS =>?
2798  011420  000027                   CT27:   27                      ;TEST #
2799  011422  011432                           CT30                    ;NEXT TEST
2800  011424  104016                           TYPLN3                  ;TYPE LINE
2801  011426  016734                           C75
2802  011430  104012                           CHAIN                   ;CHAIN.
2803                                    ;TYPE LINE OF CHARACTERS @[\
2804  011432  000030                   CT30:   30                      ;TEST #
2805  011434  011444                           CT31                    ;NEXT TEST
2806  011436  104016                           TYPLN3                  ;TYPE LINE
2807  011440  016737                           C100
2808  011442  104012                           CHAIN                   ;CHAIN
2809                                    ;TYPE LINE OF CHARACTERS ]↑AND LEFT ARROW
2810  011444  000031                   CT31:   31                      ;TEST #
2811  011446  011456                           CT32                    ;NEXT TEST
2812  011450  104016                           TYPLN3                  ;TYPE LINE
2813  011452  016742                           C135
2814  011454  104012                           CHAIN                   ;CHAIN
2815                                    ;TYPE 2 LINES OF ALL CHARACTERS, FIRST LINE FULL SPEED. SECOND LINE WITH STALLS.
2816  011456  000032                   CT32:   32                      ;TEST #
2817  011460  011514                           CT33                    ;NEXT TEST.
2818  011462  004767  174064                   JSR     %7,FBALL        ;FILL BUFFER WITH ALL CHARACTERS.
2819  011466  042767  040000  167462           BIC     #BIT14,PRGID    ;CLEAR STALL BIT IN PRGID
2820  011474  004767  173650                   JSR     %7,TYPLN        ;TYPE LINE.
2821  011500  052767  040000  167450           BIS     #BIT14,PRGID    ;SET STALL BIT IN PRGID
2822  011506  004767  173636                   JSR     %7,TYPLN        ;TYPE LINE.
2823  011512  104012                           CHAIN                   ;CHAIN.
```

```
 2824
 2825                                          ;TYPE 12 LINES OF ASR33 WORST CASE PATTERN. ALTERNATE LINES WITH STALLS.
 2826  011514  000033                 CT33:    33                        ;TEST #
 2827  011516  011576                          CT34                      ;NEXT TEST
 2828  011520  104013                          CK33                      ;33?
 2829  011522  104012                          CHAIN                     ;NO. BYPASS TEST.
 2830  011524  104000                          TYPE                      ;TYPE "WORST CASE PATTERN TEST"
 2831  011526  017212                          WCPTST
 2832  011530  004767  174056               JSR       %7,FW336          ;PATTERN TO BUFFER.
 2833  011534  012767  000006  167526       MOV       #6,CTRA           ;SET COUNT TO 6
 2834  011542  042767  040000  167406 CT33A: BIC       #BIT14,PRGID      ;CLEAR STALL BIT IN PRGID.
 2835  011550  004767  173574               JSR       %7,TYPLN          ;TYPE LINE
 2836  011554  052767  040000  167374       BIS       #BIT14,PRGID      ;SET STALL BIT IN PRGID.
 2837  011562  004767  173562               JSR       %7,TYPLN          ;TYPE LINE.
 2838  011566  005367  167476               DEC       CTRA              ;DONE 6 TIMES?
 2839  011572  001363                       BNE       CT33A             ;BRANCH IF NOT 6 TIMES YET.
 2840  011574  104012                       CHAIN                       ;DONE. CHAIN.
 2841                                          ;TYPE 12 LINES OF ASR35 WORST CASE PATTERN. ALTERNATE LINES WITH STALLS.
 2842  011576  000034                 CT34:    34                        ;TEST #
 2843  011600  011660                          CT35                      ;NEXT TEST.
 2844  011602  104014                          CK35                      ;35?
 2845  011604  104012                          CHAIN                     ;NO. BYPASS TEST.
 2846  011606  104000                          TYPE                      ;TYPE "WORST CASE PATTERN TEST"
 2847  011610  017212                          WCPTST
 2848  011612  004767  174034               JSR       %7,FW356          ;PATTERN TO BUFFER.
 2849  011616  012767  000006  167444       MOV       #6,CTRA           ;SET COUNT TO 6.
 2850  011624  042767  040000  167324 CT34A: BIC       #BIT14,PRGID      ;CLEAR STALL BIT  IN PRGID.
 2851  011632  004767  173512               JSR       %7,TYPLN          ;TYPE LINE.
 2852  011636  052767  040000  167312       BIS       #BIT14,PRGID      ;SET STALL BIT IN PRGID.
 2853  011644  004767  173500               JSR       %7,TYPLN          ;TYPE LINE
 2854  011650  005367  167414               DEC       CTRA              ;DONE 6 TIMES?
 2855  011654  001363                       BNE       CT34A             ;BRANCH IF NOT 6 TIMES YET.
 2856  011656  104012                       CHAIN                       ;CHAIN.
 2857                                          ;RUN PROGRAM 11 AS A SUBROUTINE TO EXERCISE
 2858                                          ;CURSOR ADDRESSING FUNCTIONS IF THIS WAS A MONITOR
 2859                                          ;LOAD OR IF CELL MONITOR IS NON ZERO.
 2860  011660  000035                 CT35:    35                        ;TEST #
 2861  011662  177777                          -1                        ;LAST TEST
 2862  011664  005767  166152               TST       MONITR            ;IS THIS AN AUTOMATIC TEST?
 2863  011670  001404                       BEQ       CT35A             ;NO - EXIT
 2864  011672  004767  001116               JSR       %7,PRG11          ;RUN PROG. 11 TEST
 2865  011676  004767  173476               JSR       %7,TYPCLR         ;CLEAR DOUBLE BUFFER
 2866  011702  104012                 CT35A:   CHAIN                     ;DONE CHAIN
```

```
                                        ;PRG2-KEYBOARD TEST
011704  012767  011730  167230  PRG2:   MOV     #ETO,KSTART
011712  052767  000200  167236          BIS     #BIT7,PRGID
011720  104000                          TYPE
011722  017246                          KMSG1
011724  000167  170400                  JMP     SRSET
                                        ;TEST THAT PRESSING KEY SETS DONE FLAG.
011730  000000                  ETO:    0                               ;TEST #
011732  012034                          ET1                             ;NEXT TEST.
011734  012767  000005  167326          MOV     #5,CTRA
011742  104006                  ETOA:   STRDRV
011744  012000                          ETOB
011746  104000                          TYPE                            ;TYPE "PRESS A KEY WITHIN 10 SECS."
011750  017264                          KMSG2
011752  052777  000100  167136          BIS     #BIT6,@TKS              ;ENABLE KYBD INTERRUPT.
011760  005067  166012                  CLR     PSW
011764  104400                          DELAY                           ;WAIT 10 SECONDS
011766  023420                          10000.
011770  104000                          TYPE                            ;TYPE "NO KEYBOARD REQUEST."
011772  017466                          KMSG6
011774  104010                          EHALT                           ;HALT.
011776  000411                          BR      ETOCA
012000  105777  167112          ETOB:   TSTB    @TKS                    ;TEST FOR DONE BIT ON
012004  100403                          BMI     ETOC                    ;BRANCH IF DONE BIT SET.
012006  104000                          TYPE                            ;DONE BIT NOT SET. TYPE:FALSE  KEY-
012010  017514                          KMSG7                           ;BOARD OR READER INTERRUPT.
012012  104010                          EHALT                           ;HALT
012014  012716  012022          ETOC:   MOV     #ETOCA,@%6
012020  000002                          RTI                             ;EXIT INTERRUPT.
012022  104011                  ETOCA:  SRESET
012024  005367  167240                  DEC     CTRA                    ;DONE 5 TIMES?
012030  001344                          BNE     ETOA                    ;BRANCH IF NOT DONE.
012032  104012                          CHAIN                           ;CHAIN
                                        ;ECHO TEST. KEYED CHARACTER IS TYPED. RUBOUT ENDS ROUTINE.
012034  000001                  ET1:    1                               ;TEST #
012036  012114                          ET2                             ;NEXT TEST.
012040  104000                          TYPE                            ;TYPE TITLE AND INSTRUCTIONS.
012042  017324                          KMSG3
012044  105777  167046          ET1A:   TSTB    @TKS                    ;WAIT FOR DONE FLAG
012050  100375                          BPL     .-4
012052  117767  167042  167176          MOVB    @TKB,CRBUF              ;MOVE KYBD CHAR TO CRBUF.
012060  116777  167172  167036          MOVB    CRBUF,@TPB              ;ECHO CHAR READ.
012066  105777  167030                  TSTB    @TPS                    ;WAIT FOR PRINTER DONE.
012072  100375                          BPL     .-4
012074  042767  000200  167154          BIC     #BIT7,CRBUF             ;CLEAR BIT 7 FROM CRBUF.
012102  122767  000177  167146          CMPB    #177,CRBUF              ;COMPARE CRBUF TO RUBOUT (177)
012110  001355                          BNE     ET1A                    ;BRANCH IF NOT RUBOUT (177)
012112  104012                          CHAIN                           ;CHAIN
```

```
2916
2917                                    ;OCTAL EQUIVALENT TEST. THE OCTAL EQUIVALENT OF ANY CHARACTER KEYED
2918                                    ;IS PRINTED. RUBOUT ENDS ROUTINE.
2919  012114  000002                    ET2:    2                               ;TEST #
2920  012116  177777                            -1                              ;LAST TEST
2921  012120  104001                            TYPES                           ;TYPE TITLE AND INSTRUCTIONS.
2922  012122  017426                            KMSG4
2923  012124  017337                            KMSG3A
2924  012126  177777                            -1
2925  012130  005067  167122                    CLR     CRBUF
2926  012134  105777  166756            ET2A:  TSTB    @#TKS                   ;WAIT FOR DONE FLAG.
2927  012140  100375                            BPL     .-4
2928  012142  117767  166752  167106          MOVB    @#TKB,CRBUF             ;CHARACTER TO CRBUF
2929  012150  004567  172670                    JSR     %5,ACNV4                ;CONVERT CHAR IN CRBUF TO
2930  012154  001256                            CRBUF                           ;PRINTABLE OCTAL
2931  012156  017460                            OCTEQV
2932  012160  104000                            TYPE                            ;TYPE OCTAL EQUIVALENT
2933  012162  017456                            KMSG5
2934  012164  042767  000200  167064          BIC     #BIT7,CRBUF             ;CLEAR BIT 7 FROM CRBUF
2935  012172  022767  000177  167056          CMP     #177,CRBUF              ;TEST FOR RUBOUT CHARACTER.
2936  012200  001355                            BNE     ET2A                    ;BRANCH IF NOT RUBOUT (177).
2937  012202  104012                            CHAIN                           ;CHAIN.
```

```
2938
2939                                    ;PRG3-PRINTER EXERCISER. KEYBOARD CONTROLLED.
2940                                    ;TYPES LINES WITH ANY 5 CHARACTERS. STALLS OR FULL SPEED.
2941  012204  004767  173260      PRG3:  JSR     %7,STBF          ;SET UP BUFFER.
2942  012210  104000                     TYPE                    ;TYPE TITLE
2943  012212  017542                     P7MG1
2944  012214  052767  040000  166734 HTA:  BIS     #BIT14,PRGID     ;SET STALL BIT IN PRGID.
2945  012222  012767  177600  172366      MOV     #177600,STLMSK   ;SET STALL MASK
2946  012230  012703  017623             MOV     #BLOCK1,%3
2947  012234  104000                     TYPE                    ;TYPE "TYPE IN DATA".
2948  012236  017570                     P7MG2
2949  012240  012767  000006  167022      MOV     #6,CTRA          ;CHAR COUNT TO CTRA.
2950  012246  004767  000060      HTB:   JSR     %7,GKBCR         ;GET AND STORE KYBD CHARACTER.
2951  012252  005367  167012             DEC     CTRA             ;GOT 6 CHARACTERS?
2952  012256  001373                     BNE     HTB              ;BRANCH IF NOT 6 CHARS YET.
2953  012260  042767  000200  166770      BIC     #BIT7,CRBUF
2954  012266  122767  000177  166762      CMPB    #177,CRBUF       ;CHECK 6TH CHAR FOR RUBOUT.
2955  012274  001003                     BNE     HTC              ;BRANCH IF NOT A RUBOUT.
2956  012276  042767  040000  166652      BIC     #BIT14,PRGID     ;RUBOUT. CLEAR STALL BIT IN PRGID.
2957  012304  004567  172646      HTC:   JSR     %5,BMOVE         ;FILL 72 CHAR LINE.
2958  012310  017623                     BLOCK1
2959  012312  017630                     BLOCK1+5
2960  012314  000103                     67.
2961  012316  004767  173026      HTD:   JSR     %7,TYPLN         ;TYPE LINE.
2962  012322  005777  166560             TST     @SR              ;CHANGE DATA? (SR15=1).
2963  012326  100732                     BMI     HTA              ;YES. GO CHANGE DATA
2964  012330  000772                     BR      HTD              ;NO CONTINUE WITH SAME DATA.
2965  012332  105777  166560      GKBCR: TSTB    @TKS             ;WAIT FOR DONE FLAG.
2966  012336  100375                     BPL     .-4
2967  012340  117767  166554  166710      MOVB    @TKB,CRBUF       ;CHARACTER TO CRBUF.
2968  012346  116713  166704             MOVB    CRBUF,(3)        ;CHARACTER TO LINE BUFFER.
2969  012352  142723  000200             BICB    #200,(3)+
2970  012356  116700  166674             MOVB    CRBUF,%0
2971  012362  004767  172626             JSR     %7,LSPCH         ;ECHO CHARACTER.
2972  012366  000207                     RTS     %7
```

```
                                        ;
                                        ;
                                        ;PRG4-PUNCH CLOCK ADJUSTMENT ROUTINE.
                                        ;OUTPUTS CHARACTER SET IN LEFT HALF OF SR, AND
                                        ;STALLS FOR NUMBER OF MILLISECONDS SET IN RIGHT HALF OF SR.
012370  104005                  PRG4:   CHALT                   ;HALT TO SET SR.
012372  004767  000036          ITA:    JSR     %7,C1112        ;GO OUTPUT CHARACTER SET IN LEFT
012376  000775                          BR      ITA             ;HALF OF SR AND STALL PER SR RIGHT.
                                        ;
                                        ;
                                        ;PRG5-READER CLOCK ADJUSTMENT ROUTINE.
                                        ;PERFORMS SAME FUNCTION AS PRG11, AND IN ADDITION,
                                        ;USING THE PUNCH MAINTENANCE BIT, SHIFTS OUTPUT OF PUNCH
                                        ;SHIFT REGISTER ONTO THE READER BUFFER. THE CONTENTS OF THE
                                        ;READER BUFFER ARE THEN "FIXED" ON THE CONSOLE DATA LIGHTS
                                        ;BY ISSUING A RESET WITH CONTENTS OF READER BUFFER LOADED IN R0.
012400  104005                  PRG5:   CHALT                   ;HALT TO SET SR.
012402  004767  000020          JTA:    JSR     %7,C1112M       ;GO OUTPUT CHARACTER FROM SR LEFT AND
012406  017700  166506                  MOV     @TKB,%0         ;STALL PER SR RIGHT. (TKB) TO R0.
012412  000005                          RESET                   ;"FIX" (TKB) IN DATA LIGHTS.
012414  000005                          RESET
012416  000005                          RESET
012420  000005                          RESET
012422  000005                          RESET
012424  000766                          BR      JTA             ;REPEAT.
                                        ;
012426  052777  000004  166466  C1112M: BIS     #4,@TPS         ;SET MAINTENANCE MODE (PUNCH).
012434  117767  166446  000022  C1112:  MOVB    @SR,XTY         ;STALL COUNT TO XTY.
012442  005767  000016                  TST     XTY             ;DISREGARD 0 DELAY.
012446  001002                          BNE     C1112A
012450  005267  000010                  INC     XTY
012454  117777  166427  166442  C1112A: MOVB    @SR+1,@TPB      ;LOAD PUNCH BUFFER.
012462  104400                          DELAY                   ;DELAY (APPROXIMATELY) THE NUMBER OF
012464  000000                  XTY:    OPEN                    ;MSECS. SPECIFIED AT SR RIGHT
012466  000207                          RTS     %7              ;EXIT
```

```
3008
3009                                    ;PRG6-MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
3010                                    ;WITH MAINTENANCE MODE SET, OUTPUTS ONTO PUNCH BUFFER AND BACK ONTO
3011                                    ;READER BUFFER THE CHARACTER SET IN SR LEFT. THE CHARACTER IN THE
3012                                    ;READER BUFFER IS COMPARED TO THE CHARACTER IN SR LEFT. IF THE 2 CHARACTERS
3013                                    ;DISAGREE THE PROGRAM HALTS. THE DATA LIGHTS WILL THEN CONTAIN:
3014                                    ;
3015                                    ;LEFT HALF: THE EXPECTED CHARACTER (SR LEFT).
3016                                    ;RIGHT HALF: THE CHARACTER IN THE READER BUFFER.
3017  012470  104005                    PRG6:   CHALT                   ;HALT TO SET SR.
3018  012472  052777  000004  166422    KTA:    BIS     #4,@TPS         ;SET MAINTENANCE MODE.
3019  012500  105777  166416            KTB:    TSTB    @TPS            ;WAIT FOR READY.
3020  012504  100375                            BPL     .-4
3021  012506  117767  166375  166543            MOVB    @SR+1,CRBUF+1   ;S/B CHAR TO CRBUF+1.
3022  012514  116777  166537  166402            MOVB    CRBUF+1,@TPB    ;OUTPUT CHARACTER.
3023  012522  105777  166370                    TSTB    @TKS            ;WAIT FOR READER DONE FLAG.
3024  012526  100375                            BPL     .-4
3025  012530  117767  166364  166520            MOVB    @TKB,CRBUF      ;CHAR READ TO CRBUF.
3026  012536  104004                            DATCHK                  ;GO CHECK AGAINST S/B CHAR.
3027  012540  000754                            BR      KTA             ;REPEAT.
3028
3029                                    ;PRG7-MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST.
3030                                    ;PERFORMS SAME OPERATION AS PRG6, EXCEPT THAT SPECIAL BINARY COUNT
3031                                    ;PATTERN IS USED.
3032  012542  004767  172076            PRG7:   JSR     %7,INBIN        ;INITIALIZE BINARY COUNT
3033  012546  012767  177600  172042            MOV     #177600,STLMSK  ;SET STALL LIMIT
3034  012554  052767  040000  166374            BIS     #BIT14,PRGID    ;ALLOW STALLS
3035  012562  052777  000004  166332    LTA:    BIS     #4,@TPS         ;SET MAINTENANCE MODE.
3036  012570  032777  000400  166310            BIT     #BIT8,@SR       ;CHECK STALL SWITCH
3037  012576  001001                            BNE     LTB             ;BRANCH IF NO STALL WANTED
3038  012600  104002                            STALL                   ;STALL
3039  012602  105777  166314            LTB:    TSTB    @TPS            ;WAIT FOR READY.
3040  012606  100375                            BPL     .-4
3041  012610  004767  172134                    JSR     %7,GTBINP       ;GET BIN CHARACTER.
3042  012614  110167  166437                    MOVB    %1,CRBUF+1      ;MOVE TO S/B CHAR.
3043  012620  110177  166300                    MOVB    %1,@TPB         ;OUTPUT BIN CHARACTER.
3044  012624  105777  166266                    TSTB    @TKS            ;WAIT FOR READER DONE.
3045  012630  100375                            BPL     .-4
3046  012632  117767  166262  166416            MOVB    @TKB,CRBUF      ;CHAR IN READ BUFFER TO CRBUF.
3047  012640  104004                            DATCHK                  ;GO CHECK AGAINST S/B CHAR.
3048  012642  005767  165174                    TST     MONITR          ;MONITOR LOAD?
3049  012646  001401                            BEQ     LTC             ;NO - CONTINUE
3050  012650  000207                            RTS     %7              ;YES - RETURN
3051  012652  000743            LTC:    BR      LTA             ;CONTINUE.
3052
3053                                    ;PRG10 ROLE UP TEST
3054                                    ;THE FUNCTION OF THIS TEST IS TO TEST THE ROLL-UP CAPABILITY
3055                                    ;OF THE VT05
3056                                    ;TO DO THIS A LINE OF A CHARACTER AND IT'S COMPLEMENT FOLLOWED
3057                                    ;BY A LINE OF THE COMPLEMENT AND THE CHARACTER IS TRANSMITTED
3058                                    ;THIS SCHEME IS CONTINUED UNTIL SWITCH 15 IS RAISED
3059                                    ;THE CHARACTER SHOULD NOT BE CHANGED UNTIL THE SCREEN HAS BEEN
3060                                    ;COMPLETELY FILLED
3061
3062  012654  012767  177736  000126    PRG10:  MOV     #-42,TCHAR      ;INIT TEMP CHAR
3063  012662  016767  000122  000122    RENIT:  MOV     TCHAR,CHAR      ;COMPLEMENT OF "!"
```

```
3064  012670  012767  177670  000110  PRG10C: MOV     #-72.,CNT       ;72 CHAR/LINE
3065  012676  005167  000110          PRG10D: COM     CHAR            ;
3066  012702  016700  000104          PRG10A: MOV     CHAR,%0         ;LOAD "!"
3067  012706  004767  172302                  JSR     %7,LSPCH        ;PUNCH "!"
3068  012712  005167  000074                  COM     CHAR            ;COMPLEMENT TO "↑"
3069  012716  016700  000070                  MOV     CHAR,%0         ;LOAD "↑"
3070  012722  004767  172266                  JSR     %7,LSPCH        ;PUNCH "↑"
3071  012726  005167  000060                  COM     CHAR            ; "!"
3072  012732  062767  000002  000046         ADD     #2,CNT          ;END OF LINE?
3073  012740  001360                          BNE     PRG10A          ;NO
3074  012742  012700  000015                  MOV     #15,%0          ;CR
3075  012746  004767  172242                  JSR     %7,LSPCH
3076  012752  012700  000012                  MOV     #12,%0          ;LF
3077  012756  004767  172232                  JSR     %7,LSPCH
3078  012762  005777  166120                  TST     @JSR            ;NEXT CHAR
3079  012766  100340                          BPL     PRG10C          ;NO
3080  012770  005367  000014                  DEC     TCHAR           ;YES CHANGE TCHAR
3081  012774  022767  177677  000006         CMP     #177677,TCHAR   ;CHAR STRING COMPLETE
3082  013002  001724                          BEQ     PRG10
3083  013004  000726                          BR      RENIT
3084  013006  000000                  CNT:    0
3085  013010  177736                  TCHAR:  -42
3086  013012  000041                  CHAR:   41
3087
3088                                  ;PRG11-CURSOR ADDRESS TEST
3089                                  ;   RANDOM NUMBERS ARE GENERATED AND USED AS "X" AND "Y" COORDINATES
3090                                  ;ADDRESSING A 1440 CHARACTER PRINTOUT.
3091                                  ;VERIFICATION OF DISPLAY IS PERFORMED VISUALLY BY THE USER
3092                                  ;THIS PROGRAM MAY BE RERUN AFTER COMPLETION, BY DEPRESSING THE CONT. KEY
3093
3094  013014  005067  000442          PRG11:  CLR     OVRAL
3095  013020  000167  000262                  JMP     SRT
3096  013024  012700  000035          PGO:    MOV     #35,%0          ;LOAD "CURSOR HOME"
3097  013030  004767  172160                  JSR     %7,LSPCH        ;PRINT (MOVE) CURSOR "HOME"
3098  013034  012700  000037                  MOV     #37,%0          ;LOAD "PAGE ERASE"
3099  013040  004767  172150                  JSR     %7,LSPCH        ;ERASE CRT FACE
3100  013044  016767  000414  000406  GENER: MOV     RTWY1,SET       ;SETUP PRINTOUT
3101  013052  004767  171006                  JSR     %7,RNGEN        ;GENERATE RANDOM NUMBER
3102  013056  005100                          COM     %0
3103  013060  042700  177700                  BIC     #177700,%0      ;RANDOM NO. MUST BE TWO DIGITS
3104  013064  020027  000037                  CMP     %0,#37          ;NO. MUST BE LESS THAN 40
3105  013070  101765                          BLOS    GENER           ;LOWER, REGENERATION
3106  013072  020067  000352                  CMP     %0,HYCOR        ;NO. MUST NOT BE GREATER THAN 63
3107  013076  101362                          BHI     GENER           ;GREATER, REGENERATION
3108  013100  010067  000346                  MOV     %0,YADDS        ;STORE RANDOM Y COORDINATE
3109  013104  010001                          MOV     %0,%1           ;STORE Y COORDINATE IN HI BYTE OF BUFFER
3110  013106  166701  000334                  SUB     LYCOR,%1        ;MINIMUM X,Y COORDINATE
3111  013112  001405                          BEQ     GENRX           ;RESULT, MINIMUM Y COORDINATE
3112  013114  062767  000110  000336  RATZ:   ADD     #110,SET        ;SETUP MINIMUM Y LOCATION FOR PRINTOUT
3113  013122  005301                          DEC     %1
3114  013124  001373                          BNE     RATZ            ;Y COORDINATE IS SET
3115  013126  004767  170732          GENRX:  JSR     %7,RNGEN        ;GENERATE RANDOM NUMBER
3116  013132  005100                          COM     %0
3117  013134  042700  177600                  BIC     #177600,%0      ;RANDOM NO. MAY BE LESS THAN 200
3118  013140  020027  000037                  CMP     %0,#37          ;NO. MUST NOT BE LESS THAN 40
3119  013144  101770                          BLOS    GENRX           ;LOWER, REGENERATION
```

```
3120  013146  020067  000302                  CMP     %0,HXCOR        ;NO. MUST NOT BE GREATER THAN 147
3121  013152  101365                          BHI     GENRX           ;GREATER, REGENERATION
3122  013154  010067  000276                  MOV     %0,XADDS        ;STORE RANDOM X COORDINATE
3123  013160  166700  000262                  SUB     LYCOR,%0        ;SETUP MINIMUM LOCATION
3124  013164  060067  000270                  ADD     %0,SET          ;SETUP X COOR. FOR PNTOUT.
3125  013170  016701  000264                  MOV     SET,%1          ;SETUP CHECK
3126  013174  105711                          TSTB    (1)             ;HAS CURRENT CHAR. ALREADY BEEN USED?
3127  013176  100502                          BMI     TROB            ;YES, INCREMENT RANDOM PATTERNS
3128  013200  012700  000016                  MOV     #16,%0          ;LOAD CURSOR ADDRESSING CODE
3129  013204  004767  172004                  JSR     %7,LSPCH        ;PRINT CAD
3130  013210  016700  000236                  MOV     YADDS,%0        ;LOAD Y COORDINATE
3131  013214  004767  171774                  JSR     %7,LSPCH        ;PRINT (MOVE) CURSOR TO Y COORDINATE
3132  013220  016700  000232                  MOV     XADDS,%0        ;LOAD X COORDINATE
3133  013224  004767  171764                  JSR     %7,LSPCH        ;PRINT (MOVE) CURSOR TO X COORDINATE
3134  013230  111100                          MOVB    (1),%0          ;LOAD CHARACTER TO BE PRINTED
3135  013232  004767  171756                  JSR     %7,LSPCH        ;PRINT CHARACTER
3136  013236  152711  000200                  BISB    #200,(1)        ;INDICATE USE OF CURSOR POSITION
3137  013242  005367  000214                  DEC     OVRAL           ;MAXIMUM NO. OF COORDINATES
3138  013246  001402                          BEQ     RESRT           ;IS CRT COMPLETE?
3139  013250  000167  177570                  JMP     GENER           ;NO, GENERATE ANOTHER NUMBER
3140  013254  005767  164562          RESRT:  TST     MONITR          ;AUTOMATIC TESTING DESIRED?
3141  013260  001401                          BEQ     RESRT1          ;NO - HALT
3142  013262  000207                          RTS     %7              ;YES - RETURN
3143  013264  000000                  RESRT1: HALT                    ;WAIT FOR CONTINUE
3144  013266  012767  001233  170636          MOV     #1233,RP1       ;SETUP RANDOM GENERATOR
3145  013274  012767  005622  170632          MOV     #5622,RP2
3146  013302  000167  177506                  JMP     PRG11
3147  013306  016703  000152          SRT:    MOV     RTWY1,%3        ;SETUP BUFFER FOR REFRESHING
3148  013312  142723  000200          SRT1:   BICB    #200,(3)+       ;REFRESH BUFFER
3149  013316  005267  000140                  INC     OVRAL           ;MAXIMUM NO. OF COORDINATES
3150  013322  026727  000134  002640          CMP     OVRAL,#2640     ;RESET MAX. NO OF CHARACTERS
3151  013330  001402                          BEQ     WT              ;ONCE RESET, WAIT FOR VISUAL INSP.
3152  013332  000167  177754                  JMP     SRT1            ;IF NOT RESET, CONTINUE
3153  013336  012767  000010  000074  WT:     MOV     #10,TWW         ;RESET COUNTERS
3154  013344  012767  000001  000070          MOV     #1,STEP
3155  013352  012767  000001  000064          MOV     #1,STEP1
3156  013360  000167  177440                  JMP     PGO             ;RESTART
3157  013364  016767  000052  170540  NUM:    MOV     STEP,RP1        ;USED BY RANDOM NO. GENERATOR
3158  013372  016767  000046  170534          MOV     STEP1,RP2
3159  013400  000167  177440                  JMP     GENER
3160  013404  005367  000030          TROB:   DEC     TWW             ;NOS. BEING GENERATED HAVE BEEN USED
3161  013410  001402                          BEQ     SAD
3162  013412  000167  177426                  JMP     GENER
3163  013416  012767  000010  000014  SAD:    MOV     #10,TWW         ;CHANGE RANDOM PATTERNS
3164  013424  005267  000012                  INC     STEP
3165  013430  005267  000010                  INC     STEP1
3166  013434  000167  177724                  JMP     NUM
3167
3168  013440  000010                  TWW:    10
3169  013442  000001                  STEP:   1
3170  013444  000001                  STEP1:  1
3171  013446  000040                  LYCOR:  40
3172  013450  000063                  HYCOR:  63
3173  013452  000000                  YADDS:  OPEN
3174  013454  000147                  HXCOR:  147
3175  013456  000000                  XADDS:  OPEN
```

```
  3176  013460  000000                     SET:     OPEN
  3177  013462  000000                     OVRAL:   OPEN
  3178  013464  013500                     RTWY1:   RTWY
  3179  013466  004777  164350             LOGICAL:          JSR     %7,@MONITR
  3180  013472  000240                              NOP
  3181  013474  000240                              NOP
  3182  013476  000240                              NOP
  3183  013500  042074  043511  052111     RTWY:    .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>"""""""""""""""""""""""""""""""""""""""""
  3184  013506  046101  042452  052521
  3185  013514  050111  042515  052116
  3186  013522  041452  051117  047520
  3187  013530  040522  044524  047117
  3188  013536  021076  021042  021042
  3189  013544  021042  021042  021042
  3190  013552  021042  021042  021042
  3191  013560  021042  021042  021042
  3192  013566  021042  021042  021042
  3193  013574  021042  021042  021042
  3194  013602  021042  021042  021042
  3195  013610  042074  043511  052111              .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>#########################################
  3196  013616  046101  042452  052521
  3197  013624  050111  042515  052116
  3198  013632  041452  051117  047520
  3199  013640  040522  044524  047117
  3200  013646  021476  021443  021443
  3201  013654  021443  021443  021443
  3202  013662  021443  021443  021443
  3203  013670  021443  021443  021443
  3204  013676  021443  021443  021443
  3205  013704  021443  021443  021443
  3206  013712  021443  021443  021443
  3207  013720  042074  043511  052111              .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  3208  013726  046101  042452  052521
  3209  013734  050111  042515  052116
  3210  013742  041452  051117  047520
  3211  013750  040522  044524  047117
  3212  013756  022076  022044  022044
  3213  013764  022044  022044  022044
  3214  013772  022044  022044  022044
  3215  014000  022044  022044  022044
  3216  014006  022044  022044  022044
  3217  014014  022044  022044  022044
  3218  014022  022044  022044  022044
  3219  014030  042074  043511  052111              .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  3220  014036  046101  042452  052521
  3221  014044  050111  042515  052116
  3222  014052  041452  051117  047520
  3223  014060  040522  044524  047117
  3224  014066  022476  022445  022445
  3225  014074  022445  022445  022445
  3226  014102  022445  022445  022445
  3227  014110  022445  022445  022445
  3228  014116  022445  022445  022445
  3229  014124  022445  022445  022445
  3230  014132  022445  022445  022445
  3231  014140  042074  043511  052111              .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

```
3232  014146  046101  042452  052521
3233  014154  050111  042515  052116
3234  014162  041452  051117  047520
3235  014170  040522  044524  047117
3236  014176  023076  023046  023046
3237  014204  023046  023046  023046
3238  014212  023046  023046  023046
3239  014220  023046  023046  023046
3240  014226  023046  023046  023046
3241  014234  023046  023046  023046
3242  014242  023046  023046  023046
3243  014250  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>((((((((((((((((((((((((((((((((((((((((((((((((
3244  014256  046101  042452  052521
3245  014264  050111  042515  052116
3246  014272  041452  051117  047520
3247  014300  040522  044524  047117
3248  014306  024076  024050  024050
3249  014314  024050  024050  024050
3250  014322  024050  024050  024050
3251  014330  024050  024050  024050
3252  014336  024050  024050  024050
3253  014344  024050  024050  024050
3254  014352  024050  024050  024050
3255  014360  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>))))))))))))))))))))))))))))))))))))))))))))))))
3256  014366  046101  042452  052521
3257  014374  050111  042515  052116
3258  014402  041452  051117  047520
3259  014410  040522  044524  047117
3260  014416  024476  024451  024451
3261  014424  024451  024451  024451
3262  014432  024451  024451  024451
3263  014440  024451  024451  024451
3264  014446  024451  024451  024451
3265  014454  024451  024451  024451
3266  014462  024451  024451  024451
3267  014470  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>****************************************
3268  014476  046101  042452  052521
3269  014504  050111  042515  052116
3270  014512  041452  051117  047520
3271  014520  040522  044524  047117
3272  014526  025076  025052  025052
3273  014534  025052  025052  025052
3274  014542  025052  025052  025052
3275  014550  025052  025052  025052
3276  014556  025052  025052  025052
3277  014564  025052  025052  025052
3278  014572  025052  025052  025052
3279  014600  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>++++++++++++++++++++++++++++++++++++++++
3280  014606  046101  042452  052521
3281  014614  050111  042515  052116
3282  014622  041452  051117  047520
3283  014630  040522  044524  047117
3284  014636  025476  025453  025453
3285  014644  025453  025453  025453
3286  014652  025453  025453  025453
3287  014660  025453  025453  025453
```

```
3288  014666  025453  025453  025453
3289  014674  025453  025453  025453
3290  014702  025453  025453  025453
3291  014710  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>=======================================
3292  014716  046101  042452  052521
3293  014724  050111  042515  052116
3294  014732  041452  051117  047520
3295  014740  040522  044524  047117
3296  014746  036476  036475  036475
3297  014754  036475  036475  036475
3298  014762  036475  036475  036475
3299  014770  036475  036475  036475
3300  014776  036475  036475  036475
3301  015004  036475  036475  036475
3302  015012  036475  036475  036475
3303  015020  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>.....................................
3304  015026  046101  042452  052521
3305  015034  050111  042515  052116
3306  015042  041452  051117  047520
3307  015050  040522  044524  047117
3308  015056  027076  027056  027056
3309  015064  027056  027056  027056
3310  015072  027056  027056  027056
3311  015100  027056  027056  027056
3312  015106  027056  027056  027056
3313  015114  027056  027056  027056
3314  015122  027056  027056  027056
3315  015130  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>/////////////////////////////////////////
3316  015136  046101  042452  052521
3317  015144  050111  042515  052116
3318  015152  041452  051117  047520
3319  015160  040522  044524  047117
3320  015166  027476  027457  027457
3321  015174  027457  027457  027457
3322  015202  027457  027457  027457
3323  015210  027457  027457  027457
3324  015216  027457  027457  027457
3325  015224  027457  027457  027457
3326  015232  027457  027457  027457
3327  015240  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
3328  015246  046101  042452  052521
3329  015254  050111  042515  052116
3330  015262  041452  051117  047520
3331  015270  040522  044524  047117
3332  015276  030076  030060  030060
3333  015304  030060  030060  030060
3334  015312  030060  030060  030060
3335  015320  030060  030060  030060
3336  015326  030060  030060  030060
3337  015334  030060  030060  030060
3338  015342  030060  030060  030060
3339  015350  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>111111111111111111111111111111111111111
3340  015356  046101  042452  052521
3341  015364  050111  042515  052116
3342  015372  041452  051117  047520
3343  015400  040522  044524  047117
```

```
3344  015406  030476  030461  030461
3345  015414  030461  030461  030461
3346  015422  030461  030461  030461
3347  015430  030461  030461  030461
3348  015436  030461  030461  030461
3349  015444  030461  030461  030461
3350  015452  030461  030461  030461
3351  015460  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>2222222222222222222222222222222222222222
3352  015466  046101  042452  052521
3353  015474  050111  042515  052116
3354  015502  041452  051117  047520
3355  015510  040522  044524  047117
3356  015516  031076  031062  031062
3357  015524  031062  031062  031062
3358  015532  031062  031062  031062
3359  015540  031062  031062  031062
3360  015546  031062  031062  031062
3361  015554  031062  031062  031062
3362  015562  031062  031062  031062
3363  015570  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>3333333333333333333333333333333333333333
3364  015576  046101  042452  052521
3365  015604  050111  042515  052116
3366  015612  041452  051117  047520
3367  015620  040522  044524  047117
3368  015626  031476  031463  031463
2369  015634  031463  031463  031463
3370  015642  031463  031463  031463
3371  015650  031463  031463  031463
3372  015656  031463  031463  031463
3373  015664  031463  031463  031463
3374  015672  031463  031463  031463
3375  015700  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>4444444444444444444444444444444444444444
3376  015706  046101  042452  052521
3377  015714  050111  042515  052116
3378  015722  041452  051117  047520
3379  015730  040522  044524  047117
3380  015736  032076  032064  032064
3381  015744  032064  032064  032064
3382  015752  032064  032064  032064
3383  015760  032064  032064  032064
3384  015766  032064  032064  032064
3385  015774  032064  032064  032064
3386  016002  032064  032064  032064
3387  016010  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>5555555555555555555555555555555555555555
3388  016016  046101  042452  052521
3389  016024  050111  042515  052116
3390  016032  041452  051117  047520
3391  016040  040522  044524  047117
3392  016046  032476  032465  032465
3393  016054  032465  032465  032465
3394  016062  032465  032465  032465
3395  016070  032465  032465  032465
3396  016076  032465  032465  032465
3397  016104  032465  032465  032465
3398  016112  032465  032465  032465
3399  016120  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>6666666666666666666666666666666666666666
```

```
3400  016126  046101  042452  052521
3401  016134  050111  042515  052116
3402  016142  041452  051117  047520
3403  016150  040522  044524  047117
3404  016156  033076  033066  033066
3405  016164  033066  033066  033066
3406  016172  033066  033066  033066
3407  016200  033066  033066  033066
3408  016206  033066  033066  033066
3409  016214  033066  033066  033066
3410  016222  033066  033066  033066
3411  016230  042074  043511  052111          .ASCII  '<DIGITAL*EQUIPMENT*CORPORATION>7777777777777777777777777777777777777    '
3412  016236  046101  042452  052521
3413  016244  050111  042515  052116
3414  016252  041452  051117  047520                                                                                                  '
3415  016260  040522  044524  047117
3416  016266  033476  033467  033467
3417  016274  033467  033467  033467
3418  016302  033467  033467  033467
3419  016310  033467  033467  033467
3420  016316  033467  033467  033467
3421  016324  033467  033467  033467
3422  016332  033467  033467  033467
3423  016340  021445  052040  051505  DONE:    .ASCII  '%# TEST COMPLETED@'
3424  016346  020124  047503  050115
3425  016354  042514  042524  040104
3426  016362  021445  051440  044527  ERRORM:  .ASCII  '%# SWITCH OR CONSOLE ERROR DEVICE '
3427  016370  041524  020110  051117
3428  016376  041440  047117  047523
3429  016404  042514  042440  051122
3430  016412  051117  042040  053105
3431  016420  041511  020105
3432  016424  033461  032466  030060  ERRORA:  .ASCII  '176500 PROGRAM ADDRESS '
3433  016432  050040  047522  051107
3434  016440  046501  040440  042104
3435  016446  042522  051523     040
3436  016453     060  030060  030060  ERRORB:  .ASCII  '000000@ '
3437  016460  040060     040
3438  016463     045  020043  051105  CERR:    .ASCII  '%# ERROR DEVICE '
3439  016470  047522  020122  042504
3440  016476  044526  042503     040
3441  016503     061  033067  030065  CERRA:   .ASCII  '176500 PROGRAM ADDRESS '
3442  016510  020060  051120  043517
3443  016516  040522  020115  042101
3444  016524  051104  051505  020123
3445  016532  030060  030060  030060  CERRB:   .ASCII  '000000@'
3446  016540     100
3447  016541     045  020043  040504  DERR:    .ASCII  '%# DATA ERROR DEVICE '
3448  016546  040524  042440  051122
3449  016554  051117  042040  053105
3450  016562  041511  020105
3451  016566  033461  032466  030060  DERRA:   .ASCII  '176500@'
3452  016574     100
3453  016575     045  020043  053523  ABORT:   .ASCII  '%# SWITCH ERROR TEST ABORTED@'
3454  016602  052111  044103  042440
3455  016610  051122  051117  052040
```

```
3456  016616  051505  020124  041101
3457  016624  051117  042524  040104
3458
3459
3460
3461  016632      047      137      127  A33WP6: .BYTE   047,137,127,057,127,137
3462  016635      057      127      137
3463  016640      047      133      077  A35WP6: .BYTE   047,133,077,103,077,133
3464  016643      103      077      133
3465  016646      101      102      103  A:      .BYTE   101,102,103
3466  016651      104      105      106  D:      .BYTE   104,105,106
3467  016654      107      110      111  G:      .BYTE   107,110,111
3468  016657      112      113      114  J:      .BYTE   112,113,114
3469  016662      115      116      117  M:      .BYTE   115,116,117
3470  016665      120      121      122  P:      .BYTE   120,121,122
3471  016670      123      124      125  S:      .BYTE   123,124,125
3472  016673      126      127      130  V:      .BYTE   126,127,130
3473  016676      131      132      060  Y:      .BYTE   131,132,060
3474  016701      061      062      063  ONE:    .BYTE   061,062,063
3475  016704      064      065      066  FOUR:   .BYTE   064,065,066
3476  016707      067      070      071  SEVEN:  .BYTE   067,070,071
3477  016712      041      042      043  C41:    .BYTE   041,042,043
3478  016715      044      045      046  C44:    .BYTE   044,045,046
3479  016720      047      050      051  C47:    .BYTE   047,050,051
3480  016723      052      053      054  C52:    .BYTE   052,053,054
3481  016726      055      056      057  C55:    .BYTE   055,056,057
3482  016731      072      073      074  C72:    .BYTE   072,073,074
3483  016734      075      076      077  C75:    .BYTE   075,076,077
3484  016737      100      133      134  C100:   .BYTE   100,133,134
3485  016742      135      136      137  C135:   .BYTE   135,136,137
3486  016745      377      000      377  C377:   .BYTE   377,000,377
3487  016750  021445  040524  020102  TBTST:  .ASCII  '%#↑AB ↑EST%#'
3488  016756  042524  052123  021445
3489  016764  020040  020040  020040  TBMRK:  .ASCII  '        /@'
3490  016772  020040  040057
3491  016776  020040  020040  020040  TBMRK1: .ASCII  '       /@'
3492  017004  027440      100
3493  017007      045      100        CRLF:   .ASCII  '%@'
3494  017011      055  026455  044455  RM33A:  .ASCII  '----I@'
3495  017016      100
3496  017017      055  026511      100  RM33B:  .ASCII  '-I-@'
3497  017023      055  026455  044455  RM37A:  .ASCII  '----I-I@'
3498  017030  044455      100
3499  017033      134  040040        SPTSTC: .ASCII  '\ @'
3500  017036  021445  040503  051122  CRTST:  .ASCII  '%#CARRIAGE RETURN TEST%#@'
3501  017044  040511  042507  051040
3502  017052  052105  051125  020116
3503  017060  042524  052123  021445
3504  017066      100
3505  017067      045  051043  043511  RMTST:  .ASCII  '%#RIGHT MARGIN TEST%#@'
3506  017074  052110  046440  051101
3507  017102  044507  020116  042524
3508  017110  052123  021445      100
3509  017115      045  041443  051125  SPTST:  .ASCII  '%#CURSOR RIGHT TEST%#@'
3510  017122  047523  020122  044522
3511  017130  044107  020124  042524
```

```
3512  017136  052123  021445     100
3513  017143     045  046043  047111  LFTST:  .ASCII  '%#LINE FEED TEST%#@'
3514  017150  020105  042506  042105
3515  017156  052040  051505  022524
3516  017164  040043
3517  017166  021445  044103  051101  CHRTST: .ASCII  '%#CHARACTER TESTS%#@'
3518  017174  041501  042524  020122
3519  017202  042524  052123  022523
3520  017210  040043
3521  017212  021445  047527  051522  WCPTST: .ASCII  '%#WORST CASE PATTERN TEST%#@'
3522  017220  020124  040503  042523
3523  017226  050040  052101  042524
3524  017234  047122  052040  051505
3525  017242  022524  040043
3526  017246  021445  054513  042102  KMSG1:  .ASCII  '%#KYBD TEST%#@'
3527  017254  052040  051505  022524
3528  017262  040043
3529  017264  050045  042522  051523  KMSG2:  .ASCII  '%PRESS A KEY WITHIN 10 SECONDS.@'
3530  017272  040440  045440  054505
3531  017300  053440  052111  044510
3532  017306  020116  030061  051440
3533  017314  041505  047117  051504
3534  017322  040056
3535  017324  021445  041505  047510  KMSG3:  .ASCII  '%#ECHO TEST'
3536  017332  052040  051505     124
3537  017337     045  044103  051101  KMSG3A: .ASCII  '%CHARACTER KEYED WILL BE TYPED.'
3538  017344  041501  042524  020122
3539  017352  042513  042531  020104
3540  017360  044527  046114  041040
3541  017366  020105  054524  042520
3542  017374  027104
3543  017376  051045  041125  052517          .ASCII  '%RUBOUT ENDS ROUTINE.%#@'
3544  017404  020124  047105  051504
3545  017412  051040  052517  044524
3546  017420  042516  022456  040043
3547  017426  021445  041517  040524  KMSG4:  .ASCII  '%#OCTAL EQUIVALENT TEST@'
3548  017434  020114  050505  044525
3549  017442  040526  042514  052116
3550  017450  052040  051505  040124
3551  017456  020045                  KMSG5:  .ASCII  '% '
3552  017460  020040  020040  040045  OCTEQV: .ASCII  '    %@'
3553  017466  047045  020117  042513  KMSG6:  .ASCII  '%NO KEYBOARD REQUEST.@'
3554  017474  041131  040517  042122
3555  017502  051040  050505  042525
3556  017510  052123  040056
3557  017514  043045  046101  042523  KMSG7:  .ASCII  '%FALSE KYBD INTERRUPT@'
3558  017522  045440  041131  020104
3559  017530  047111  042524  051122
3560  017536  050125  040124
3561  017542  021445  044504  050123  P7MG1:  .ASCII  '%#DISPLAY EXERCISER%#@'
3562  017550  040514  020131  054105
3563  017556  051105  044503  042523
3564  017564  022522  040043
3565  017570  021445  054524  042520  P7MG2:  .ASCII  '%#TYPE IN DATA :@'
3566  017576  044440  020116  040504
3567  017604  040524  035040     100
```

```
  3568  017611      125  040040          BKSU:   .ASCII  'U @'
  3569  017614  020040  020040      040 DECVAL: .ASCII  ' ' ,
  3570  017621  000001                  DEND:   .END
```

```
A       016646
ABORT   016575
ACNV    005102
ACNVB   005036
ACNVC   005064
ACNVE   001714
ACNVM   005116
ACNVX   005100
ACNV4   005044
ACNV6   005016
ADTENP  005336
ARDA    003736
ARDB    003762
AREAD   003716
ATO     005724
ATOA    005742
ATOB    005746
ATOE    005752
AT1     005756
AT1A    005774
AT1B    006000
AT1E    006004
AT11    006224
AT11A   006234
AT11B   006244
AT12    006250
AT12A   006260
AT12B   006264
AT12C   006276
AT12D   006310
AT12E   006304
AT12F   006316
AT13    006322
AT13A   006332
AT13B   006336
AT13C   006354
AT13E   006352
AT14    006360
AT14A   006370
AT14B   006374
AT14C   006414
AT14E   006412
AT17    006420
AT17A   006430
AT17B   006434
AT17C   006454
AT17D   006502
AT17E   006476
AT17F   006510
AT2     006010
AT2A    006026
AT2B    006032
AT2E    006036
AT20    006514
AT20A   006530
AT20B   006540
AT20C   006564
AT20D   006566
AT20E   006560
AT21    006572
AT21A   006606
AT21B   006616
AT21C   006640
AT21E   006650
AT22    006656
AT22A   006702
AT22B   006736
AT22C   006740
AT22E   006732
AT23    006750
AT23A   006770
AT23B   007012
AT23C   007022
AT23D   007036
AT23E1  007010
AT23E2  007042
AT24    007050
AT24A   007066
AT24B   007110
AT24C   007130
AT24E1  007104
AT24E2  007126
AT25    007134
AT25A   007152
AT25B   007174
AT25E   007172
AT26    007200
AT26A   007210
AT26B   007232
AT26C   007252
AT26E1  007226
AT26E2  007250
AT27    007256
AT27A   007266
AT27B   007310
AT27E   007306
AT3     006042
AT3A    006060
AT3B    006064
AT3E    006070
AT30    007314
AT30A   007324
AT30B   007334
AT30E   007332
AT31    007340
AT31A   007350
AT31B   007372
AT31E   007370
AT32    007376
AT32A   007406
AT32B   007434
AT32E   007432
AT33    007440
AT33A   007450
AT33B   007474
AT33E   007472
AT34    007500
AT34A   007514
AT34B   007536
AT34C   007542
AT34E   007534
AT35    007546
AT35A   007562
AT35B   007604
AT35E   007614
AT36    007622
AT36A   007636
AT36B   007672
AT36C   007674
AT37    007704
AT37A   007714
AT37B   007742
AT37C   007752
AT37D   007766
AT37E1  007740
AT37E2  007772
AT4     006074
AT4A    006112
AT4B    006134
AT4C    006154
AT4E1   006130
AT4E2   006152
AT40    010000
AT40A   010014
AT40B   010052
AT40C   010054
AT41    010064
AT41A   010104
AT41B   010130
AT41C   010142
AT42    010152
AT42A   010166
AT42B   010214
AT42E   010212
AT43    010220
AT43A   010236
AT43B   010242
AT5     006160
AT5A    006176
AT5B    006220
AT5E    006216
A1ST    005072
A33WP6  016632
A35WP6  016640
B       002066
BDCNV   005230
BDCNVA  005250
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
BKSU    017611
BLKBB   =  017744
BLKCC   =  020056
BLK2    =  017746
BLOCKA=  017621
BLOCKB=  017733
BLOCKC=  020045
BLOCK1=  017623
BLOCK2=  017735
BMOVA   005166
BMOVE   005156
BRCTR   003764
BYPMAN=  000400
CC      =  177776
CERR    016463
CERRA   016503
CERRB   016532
CERRH   001672
CHAIN   =  104012
CHAINN  002430
CHALT   =  104005
CHAR    013012
CHKASR=  104022
CHK33   003206
CHK330  003220
CHK35   003222
CHLT    001300
CHLTR   001320
CHNA    002464
CHNAA   002512
CHNB    002526
CHRTST  017166
CHR1    001260
CHR2    001262
CHR3    001264
CKASR   003240
CK33    =  104013
CK35    =  104014
CLEAN   002736
CNT     013006
CNVCTR  005330
CONADD  001112
CONIT   003264
CONVEC  001114
CPRDY   005200
CPRDYA  005210
CRBUF   001256
CRLF    017007
CRTA    002312
CRTST   017036
CTRA    001270
CTRB    001272
CTRC    001274
CTRD    001276
CTO     010312
CTOA    010336
CTOB    010346
CTOC    010354
CT1     010414
CT1A    010440
CT10    011172
CT11    011204
CT12    011216
CT13    011230
CT14    011242
CT15    011254
CT16    011266
CT17    011300
CT2     010460
CT2A    010476
CT2B    010516
CT2C    010524
CT2D    010546
CT2E    010622
CT20    011312
CT21    011324
CT22    011336
CT23    011350
CT24    011362
CT25    011374
CT26    011406
```

```
CT27    011420
CT3     010632
CT3A    010656
CT3B    010706
CT30    011432
CT31    011444
CT32    011456
CT33    011514
CT33A   011542
CT34    011576
CT34A   011624
CT35    011660
CT35A   011702
CT4     010712
CT4A    010742
CT4B    010754
CT5     011130
CT6     011146
CT7     011160
CURADD  003552
CURTST  001144
CURVEC  003560
C100    016737
C1112   012434
C1112A  012454
C1112M  012426
C135    016742
C377    016745
C41     016712
C44     016715
C47     016720
C52     016723
C55     016726
C72     016731
C75     016734
D       016651
DATCHK= 104004
DATHLT= 104017
DECVAL  017614
DELAY = 104400
DEND    017621
DERR    016541
DERRA   016566
DERRH   001532
DIGIT   005332
DISPLA  001110
DISPRE  000174
DLCNT   004556
DLY     004514
DLYA    004534
DLYB    004540
DONE    016340
DTCHK   001436
DTCHKA  001534
DTHLT   001526
EHALT = 104010
EHLT    001322
EHLTA   001424
EHLTB   001334
EMTA    003022
EMTINT  002776
EMTTAB  001204
END     002104
ERCTR   001266
ERR     001536
ERRA    001556
ERRB    001664
ERRH    001432
ERROR = 104003
ERRORA  016424
ERRORB  016453
ERRORM  016362
ETO     011730
ETOA    011742
ETOB    012000
ETOC    012014
ETOCA   012022
ET1     012034
ET1A    012044
ET2     012114
ET2A    012134
FBALL   005552
FBF3    005506
FBF3A   005516
FBF3B   005524
FORWD   002632
FORWDA  002664
FORWDB  002672
FOUR    016704
FW336   005612
FW356   005652
G       016654
GENER   013044
GENRX   013126
GETRDY  002332
GKBCR   012332
GOTST   002710
GOTSTA  002730
GRCNT   004620
GTBIN   004702
GTBINP  004750
GTRDYA  002344
GTRDYB  002350
GTRDYC  002366
GTRDYD  002414
HLTSW = 100000
HTA     012214
HTB     012246
HTC     012304
HTD     012316
HXCOR   013454
HYCOR   013450
ICTR    001152
INBIN   004644
INCPRG  002306
INCRTN  002424
ITA     012372
J       016657
JTA     012402
KMSG1   017246
KMSG2   017264
KMSG3   017324
KMSG3A  017337
KMSG4   017426
KMSG5   017456
KMSG6   017466
KMSG7   017514
KSTART  001142
KTA     012472
KTB     012500
LFTST   017143
LOGICA  013466
LOWADD  003554
LPRGSW= 002000
LSPCH   005214
LSTADD  003556
LTA     012562
LTB     012602
LTC     012652
LYCOR   013446
M       016662
MACHER  000004
MANUAL= 100000
MONITR  000042
NITRSW= 004000
NOP   = 000240
NPRTSW= 020000
NSTART  002212
NTRCSW= 010000
NULL    004430
NUM     013364
NXTST   001150
OCTEQV  017460
ONE     016701
OPEN  = 000000
OUT     004236
OUTCAR  004356
OUTEX   004426
OUTNCT  004340
OUTNUL  004350
OUTTYP  004242
OVRAL   013462
P       016665
PC    =%000007
PFAIL   003066
PGO     013024
PIND    004674
POPSP = 005726
POPSP2= 022626
PRGEND  002624
PRGID   001156
PRGNUM  001140
PRGTAB  001160
PRGO    005712
PRG1    010260
PRG10   012654
PRG10A  012702
PRG10C  012670
PRG10D  012676
PRG11   013014
PRG2    011704
PRG3    012204
PRG4    012370
PRG5    012400
PRG6    012470
PRG7    012542
PRTYO = 000000
PRTY1 = 000040
PRTY2 = 000100
PRTY3 = 000140
PRTY4 = 000200
PRTY5 = 000240
PRTY6 = 000300
PRTY7 = 000340
PSW   = 177776
PTO     004670
PTOP    004676
PTI     004672
PTIP    004700
P7MG1   017542
P7MG2   017570
RATZ    013114
RCMSK   004640
RCNT    001254
RENIT   012662
RESRT   013254
RESRT1  013264
RESTAR  003130
RETPVT  003662
RHLTA   001710
RIND    004666
RMB     010454
RMTST   017067
RM33A   017011
RM33B   017017
RM37A   017023
RNCNT   004642
RNGEN   004064
RP1     004132
RP2     004134
RSTA    003376
RSTART  003342
RSTB    003522
RSTC    003546
RSTPC   003656
RSTPSW  003660
RSTREG= 104021
RSTRG   003622
RTNNO   001146
RTWY    013500
RTWY1   013464
R0    =%000000
R1    =%000001
R2    =%000002
R3    =%000003
R4    =%000004
R5    =%000005
S       016670
SAD     013416
SAVREG= 104020
SAVRG   003562
SAVR6   003126
SCOPSW= 040000
SCOPTR  001154
SET     013460
SEVEN   016707
SHALT   001676
SHLTA   001712
SP    =%000006
SPBOT   001104
SPCNT   011026
SPTST   017115
SPTSTC  017033
SR      001106
SRESET= 104011
SRSET   002330
SRSETT  004046
SRT     013306
SRTSW = 001000
SRT1    013312
STAL    004560
STALA   004612
STALAA  004572
STALB   004614
STALL = 104002
```

```
START   001740          SUBTNB  005310          TKB     001120          TTYTYP  001136          TYPL3   005430
STARTO  002246          SVRPC   003616          TKLVL   001130          TWW     013440          TYPS    004462
STBF    005470          SVRPSW  003620          TKS     001116          TYP     004136          TYPSA   004506
STEP    013442          SWREG   000176          TKVTR   001126          TYPA    004146          TYPSB   004510
STEP1   013444          TABP    011030          TPB     001124          TYPC    004210          V       016673
STLMSK  004616          TABPA   011042          TPBM    011000          TYPCLR  005400          WCPTST  017212
STLSPV  004016          TABPB   011052          TPBMA   011010          TYPDAT  004460          WT      013336
STLSRV  003766          TABPC   011070          TPLVL   001134          TYPE  = 104000          XADDS   013456
STNUM   003550          TBCNT   011024          TPL3A   005452          TYPES = 104001          XTY     012464
STPCHV= 104007          TBMRK   016764          TPS     001122          TYPF    004432          Y       016676
STPPA   004034          TBMRK1  016776          TPVTR   001132          TYPG    004444          YADDS   013452
STPRA   004004          TBTST   016750          TROB    013404          TYPLA   005356          $ENDAD  002614
STRDRV= 104006          TCHAR   013010          TRPA    003166          TYPLB   005362          .     = 017621


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*DZVTBD,DZVTBD/SOL=DZVTBD
RUN-TIME:  12 24 1 SECONDS
RUN-TIME RATIO:  124/38=3.2
CORE USED:  16K  (31 PAGES)
```

Spooler runtime 13 Seconds, 56 KCS, 314 disk reads, 3 disk writes, 89 pages

[redacted] Date 03-Dec-76 09:46:40 Monitor IPC-D 602 [602] [redacted]