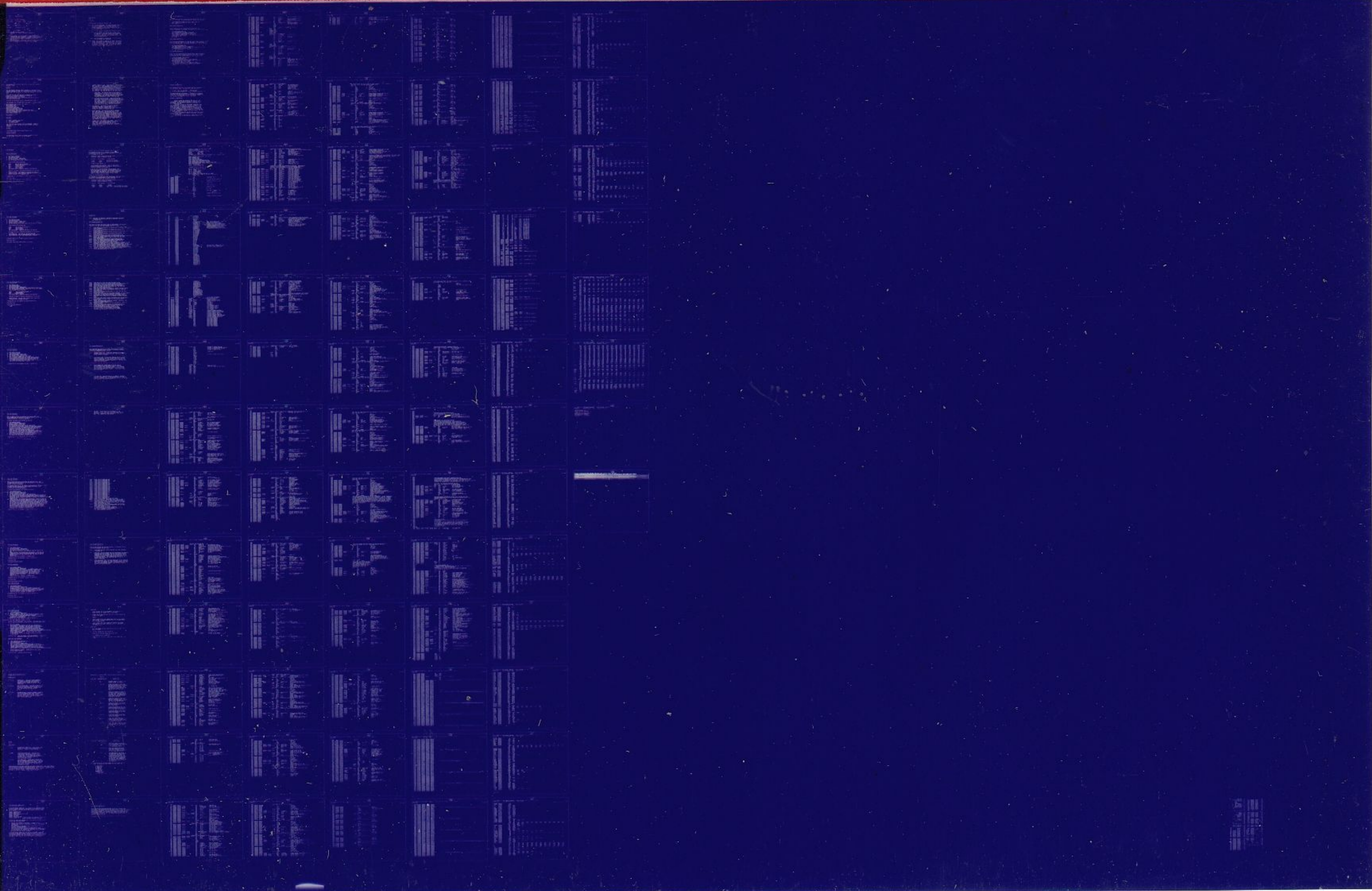


VT05

DISPLAY TEST
MD-11-DZVTB-C

EP-DZVTB-C-DL-A
COPYRIGHT 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA



IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZVTB-C
PRODUCT NAME: VTOS DISPLAY TERMINAL TESTS
DATE: JUNE, 1972
REVISED: MARCH, 1973
MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JIM CARRON

PREVIOUS NO WAS: MAINDEC-11-D5GA
COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
1973

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

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

1. ABSTRACT

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

THIS TEST CAN TEST THE VTOS 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.

THIS TEST IS COMPATIBLE FOR ALL CONFIGURATIONS OF PDP 11.

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. VTOS DISPLAY TERMINAL
- C. HIGH SPEED READER

THE VTOS MUST HAVE STANDARD TELETYPE ADDRESSES. REFER TO SECTION 7.3 IF THE VTOS 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 PROCEDURE4.1 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.

4.2 PRG1 USE PROCEDURE

- A. SET VTOS TO ON-LINE
- B. LOAD ADDRESS 000200.
- C. SET SR TO 000001. PRESS START
- D. 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
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 VTOS WILL BE EXERCISED AND THE PROGRAM WILL STOP AT PROGRAM END HALT WHEN COMPLETED.
- G. ERROR DETECTION IS BY VISUAL INSPECTION OF DISPLAY.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.2

EXECUTION TIME:

ONE NORMAL PASS TAKES APPROXIMATELY 12 MINUTES.

4.3 PRG2 USE PROCEDURE

- A. SET VTOS 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.

4.4 PR33 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

f
p
SP B D H I

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 PRGS USE PROCEDURE

PRGS 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 ADDRESS 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, OUTPUTTING 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 000007. PRESS START
- C. THE PROGRAM RUNS CONTINUOUSLY. THE SPECIAL BINARY COUNT PATTERN 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 SRB 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 CHARACTER FOLLOWED BY THE CHARACTER. THIS TEST VERIFIES THE ROLL-UP CAPABILITY OF THE VTOS.
- D. REFER TO SECTION 6. ERRORS, IF ERRORS OCCUR.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.10

EXECUTION TIME:

CONTINUOUS RUNNING PROGRAM.

4.10 PRG11 USER PROCEDURE

- A. LOAD ADDRESS 000200.
 - B. SET SR TO 000011. PRESS START
 - C. CURSOR ADDRESSING CODES WILL BE UTILIZED TO RANDOMLY COVER THE FACE OF THE CRT WITH THE APPROPRIATE MESSAGE.
AT THIS POINT THE PROGRAM WILL HALT, A RERUN IS POSSIBLE BY DEPRESSING THE CONTINUE KEY.
- A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.11

EXECUTION TIME: BAUD DEPENDANT, APPROX. 3-5 MIN.

4.11 MONITOR LOAD USE PROCEDURE - ACT11 OR DDP1 - NON PAPER TAPE SYSTEM

- A. SET SR TO 000000
- B. WHEN LOADED BY THE MONITOR THIS TEST WILL AUTOMATICALLY EXECUTE PROG. 0 AND PROG 1 FOR THE CONSOLE DEVICE AND FOR ALL CONTIGUOUSLY ASSIGNED EXTRA DEVICES. IN ADDITION PROGRAM 0 RUNS PROGRAM 7 AS A SUBROUTINE TO VERIFY THE CORRECT TRANSFER OF DATA THROUGH THE INTERFACE AND PROGRAM 1 RUNS PROGRAM 11 AS A SUBROUTINE TO VERIFY THE CURSOR ADDRESSING FEATURES
- C. REFER TO SECTION 6, ERRORS. ERRORS DETECED ARE LOGGED ON THE CONSOLE DEVICE
- D. AT THE END OF A PASS OF ALL DEVICES CONTROL IS RETURNED TO THE MONITOR

EXECUTION TIME: BAUD AND NUMBER OF DEVICES DEPENDENT. APPROXIMATELY
20 MIN FOR ERROR FREE PASS PER DEVICE

4.12 GROUP TEST USE PROCEDURE

- A. LOAD CONSTANTS AS DESCRIBED IN 7.4
- B. LOAD ADDRESS 000200
- C. SET SR TO 000000. PRESS START
- D. THIS TEST WILL AUTOMATICALLY EXECUTE PROG. 0 AND PROG 1 FOR THE GROUP OF EXTRA DEVICES AS DEFINED IN 7.4 IN ADDITION PROGRAM 0 RUNS PROGRAM 7 AS A SUBROUTINE TO VERIFY THE CORRECT TRANSFER OF DATA THROUGH THE INTERFACE AND PROGRAM 1 RUNS PROGRAM 11 AS A SUBROUTINE TO VERIFY THE CURSOR ADDRESSING FEATURES
- E. REFER TO SECTION 6 ERRORS. ERRORS DETECTED ARE LOGGED ON THE CONSOLE DEVICE.

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. ERRORS6.1 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 TESTING AN ERROR MESSAGE IS TYPED ON THE CONSOLE DEVICE INDICATING TYPE OF ERROR, ADDRESS (CSR) OF DEVICE IN ERROR AND PROGRAM ADDRESS OF TEST. NO HALT OCCURS AND THE TEST 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 OCCURRED, PERFORM THE FOLLOWING STEPS:

- A. EXAMINE THE CONTENTS OF REGISTER 6 (ADDRESS 177706).
- B. TRANSFER THE CONTENTS OF REGISTER 6 TO THE SR, LOAD ADDRESS, AND EXAMINE.
- C. THE DATA SHOWN IN THE DATA LIGHTS IS THE VALUE OF THE PC WHEN THE FAILURE OCCURRED.
- D. LOCATE IN THE PROGRAM LISTING THE DISPLAYED PC VALUE.
- E. THE INSTRUCTION THAT IMMEDIATELY PREECEDES THE ONE REFERENCED BY THE DISPLAYED PC VALUE IS THE INSTRUCTION THAT WAS 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 VT05. ANY FURTHER STEPS TO DIAGNOSE A NON-RECOVERABLE ERROR ARE NOT WITHIN THE SCOPE OF THIS PROGRAM.

7. MISCELLANEOUS7.1 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 PRGD. 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 PRG0, PRG1, AND PRG3. THESE PROGRAMS CONSIST OF A SET OF ROUTINES EACH OF WHICH SPECIFIES THE NUMBER OF TIMES A TEST IS TO BE PERFORMED BY MEANS OF AN ITERATION COUNT. SETTING SR11 TO A 1 CAUSES THE PROGRAM TO DISREGARD THE ITERATION COUNT AND PERFORM THE TEST ONLY ONCE FOR EACH ROUTINE. TWO POSSIBLE USES OF THIS OPTION ARE:

- A. QUICK PASS. THE USER MAY ELECT TO RUN THROUGH A PROGRAM QUICKLY TO FIND OUT IF ANY FAILURES SHOW IMMEDIATELY. A SUCCESSFUL QUICK PASS HOWEVER, DOES NOT GUARANTEE THAT THE SAME PROGRAM WILL RUN ERROR-FREE WHEN PERFORMING A NORMAL ITERATION PASS.
- B. SKIP OVER FAILING ROUTINE. WHEN A ROUTINE HAS DETECTED A SOLID FAILURE, THE ERROR WILL BE REPORTED MANY TIMES. TO GO ON TO THE NEXT ROUTINE, THE USER CAN INHIBIT ITERATION. IT WILL BE NECESSARY TO CAUSE THE PROGRAM TO STOP AT THE END OF THE ROUTINE BY SETTING SR15 TO A 1. OTHERWISE THE PROGRAM WOULD QUICKLY RUN THROUGH THE NEXT ROUTINE(S) ALSO.

SR10 - LOOP PROGRAM. THIS OPTION IS USED BY PROGRAMS PRG0, 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 PRG0, 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 B 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 VTOS 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 VTOS'S

THIS PROGRAM WILL AUTOMATICALLY AND CONTINUOUSLY TEST THE CONSOLE DEVICE AND A CONTIGUOUS GROUP OF VTOS'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. VTOS RECEIVER CSR ADDRESS
LSTADD	000000	LAST + 10 VTOS 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 PRGC PROGRAM DESCRIPTION

PRGC 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.

(9.1 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

S.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---

RTN2 RIGHT CURSOR TEST. CHECKS ABILITY OF THE VTDS 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.

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 {~ AND LEFT ARROW
 RTN30 TYPES LINE OF CHARACTERS | AND LEFT ARROW
 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?I
 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.

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 VTOS 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 BACKWARD 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 DOWNWARD ONE LINE. IF THE CURSOR IS ON THE BOTTOM LINE, THIS HAS NO EFFECT.

<u>OCTAL CODE</u>	<u>CORRESPONDING KEY</u>	<u>DESCRIPTION</u>
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.

8.9 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 D.
- 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 C UNDER MONITOR LOAD OR WHEN AUTOMATIC GROUP TESTING IS USED.

9.10 PROGRAM 10 DESCRIPTION

THE PURPOSE OF THIS TEST IS TO VERIFY THAT THE VTOS MEMORY HAS ROLL-UP CAPABILITIES. THE TEST FUNCTIONS AS FOLLOWS:

- A. A LINE !, AND ITS COMPLEMENT IS DISPLAYED
- B. THIS LINE IS FOLLOWED BY A LINE OF ITS EXACT COMPLEMENT

THIS PROCEDURE RUNS CONTINUOUSLY. IF SWITCH 15 IS HELD UP MOMENTARILY THE ASCII CODE FOR THE CHARACTER IS INCREMENTED BY ONE. BY UTILIZING SWITCH 15 IN THIS MANNER, PROGRAM 10 CAN TEST THE ROLL-UP CAPABILITY OF ALL CHARACTERS.

9.11 PROGRAM 11 DESCRIPTION

RANDOM NUMBERS ARE GENERATED AND USED AS X AND Y COORDINATES. A 1440 CHARACTER MESSAGE IS USED AS THE VISUAL DISPLAY, ONE PARTICULAR CHARACTER IS AVAILABLE FOR EACH CURSOR POSITION.

THE FIRST LOCATION OF THE STORED CHARACTERS (MINIMUM COORDINATES) IS USED AS A CONSTANT. A RANDOM Y COORDINATE (40 TO 63) IS GENERATED AND CONVERTED TO 0 TO 23. THIS LATER NUMBER IS THEN MULTIPLIED BY 110 WHICH PLACES US AT THE BEGINNING OF LINES 0 TO 19. AN X COORDINATE (40 TO 147) IS NOW GENERATED AND CONVERTED TO 0 TO 107 AND ADDED TO THE Y COORDINATE, ADDRESS, AND THIS CONTROLS THE TYPE OF CHARACTER OUTPUT.

AFTER 1440 CHARACTERS ARE PRINTED A HALT WILL OCCUR TO ALLOW VISUAL INSPECTION. A RERUN IS POSSIBLE BY DEPRESSING THE CONTINUE KEY.

THIS PROGRAM IS EXECUTED AS A SUBROUTINE OF PROGRAM 0 UNDER MONITOR LOAD OR WHEN AUTOMATIC GROUP TESTING IS USED.

```

.TITLE KL11 TESTS FOR VTOS DISPLAY TERMINAL
;MAINDEC 11-DZVTB-C
;KL11 TESTS FOR VTOS 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
.+2 ;UNASSIGNED TRAP
HALT
MACHERR: .+2 ;SP OVERFLOW, BUS ERROR TRAP
HALT
.+2 ;RESERVED INSTRUCTION TRAP
HALT
.+2 ;TRACE TRAP
HALT
.+2 ;TRAP TO CALL ICX
HALT
.+2 ;POWER FAIL TRAP
PRTY7
EMTINT ;EMT TRAP
PRTY7
TRPINT ;TRAP TRAP. SIMILAR TO EMT
PRTY7
MONITR: .+2 ;TRAPPED FOR MONITOR LOAD
HALT
.REPT 176
.+2
HALT
.ENDR ;TRAPPED TO PREVIOUS ADDRESS.

```

```

000000 000000
000002 000000
000004 000006
000006 000000
000010 000012
000012 000000
000014 000016
000016 000000
000020 000022
000022 000000
000024 000026
000026 000340
000030 002634
000032 000340
000034 002700
000036 000340
000040 000042
000042 000000

```

000000 000000
000002 000000
000004 000006
000006 000000
000010 000012
000012 000000
000014 000016
000016 000000
000020 000022
000022 000000
000024 000026
000026 000340
000030 002634
000032 000340
000034 002700
000036 000340
000040 000042
000042 000000

106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

177570
177776
177776
000240
000000
100000
040000
020000
010000
004000
002000
001000
000400
100000
100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000000
005726
022626
000340
000300
000240
000200
000140
000100
000040
000000
104000
104001
104002
104003
104004
104005
104006
104007
104010
104011
104012
104013

```
:EQUATE STATEMENTS
SR=177570
CC=177776
PSW=177776
NOP=240
OPEN=0
HLTSW=BIT15
SCOPSW=BIT14
NPRTSW=BIT13
NTRCSW=BIT12
NITRSW=BIT11
LPRGSW=BIT10
SRTSW=BIT9
BYPMAN=BIT8
MANUAL=BIT15
BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=0
POPSP=5726
POPSP2=022626
PTY7=340
PTY6=300
PTY5=240
PTY4=200
PTY3=140
PTY2=100
PTY1=40
PTY0=0
TYPE=EMT+0
TYPES=EMT+1
STALL=EMT+2
ERROR=EMT+3
DATCHK=EMT+4
CHALT=EMT+5
STRDRV=EMT+6
STPCHV=EMT+7
EHALT=EMT+10
SRESET=EMT+11
CHAIN=EMT+12
CK33=EMT+13
```

```
:HALT SWITCH DEFINITION
:SCOPE SWITCH DEFINITION
:INHIBIT PRINT SWITCH DEFINITION
:INHIBIT TRACE SWITCH DEFINITION
:INHIBIT ITERATION SWITCH DEFINITION
:LOOP PROGRAM SWITCH DEFINITION
:SELECT ROUTINE SWITCH DEFINITION
:BYPASS MANUAL INTERVENTION DEFINITION.
```

```
:POP THE STACK. SAME AS TST (6)+
:POP STACK TWICE. SAME AS CMP (6)+,(6)+
:PRIORITY LEVEL DEFINITIONS
```

107		
108	104014	
109	104016	
110	104017	
111	104020	
112	104021	
113	104022	
114	104400	
115	000007	
116	017441	
117	017443	
118	017553	
119	017564	
120	017555	
121	017566	
122	017665	
123	017676	
124	000200	
125	000200	001526
126	001104	
127	001106	
128	001110	
129	001112	
130	001114	
131	001116	
132	001120	
133	001122	
134	001124	
135	001126	
136	001130	
137	001132	
138	001134	
139	001136	
140	001140	
141	001142	
142	001144	
143	001146	
144	001150	
145	001152	
146	001154	
147	001156	
148	001160	
149	001162	
150	001164	
151	001166	
152	001170	
153	001172	
154	001174	
155	001176	
156		
157		

```

CK35=EMT+14
TYPLN3=EMT+16
DATHLT=EMT+17
SAVREG=EMT+20
RSTREG=EMT+21
CHKASR=EMT+22
DELAY=TRAP+0
BELL=007
BLOCKA=DEND
BLOCK1=BLOCKA+2
BLOCKB=BLOCKA+112
BLKBB=BLOCKA+123
BLOCK2=BLOCKA+114
BLK2=BLOCKA+125
BLOCKC=BLOCKA+224
BLKCC=BLOCKA+235
.=200
JMP      START      ;GO TO START OF PROGRAM.
;GET CODE OUT OF VECTOR AREA
SPBOT: 0 ;BOTTOM OF STACK
CONADD: 177560 ;CONSOLE LSR CSR POINTER
CONVEC: 60 ;CONSLE INTERRUPT VECTOR
TKS: 177560 ;LSR CSR
TKB: 177562 ;LSR BUFFER
TPS: 177564 ;LSP CSR
TPB: 177566 ;LSP BUFFER
TKVTR: 60 ;LSR INTERRUPT VECTOR
TKLVL: PRTY4 ;LSR PRIORITY LEVEL
TPVTR: 64 ;LSP INTERRUPT VECTOR
TPLVL: PRTY4 ;LSP PRIORITY LEVEL
TTYTYP: 01 ;TTY = KSR35
PRGNUM: OPEN ;CONTAINS CURRENT PROGRAM#
KSTART: OPEN ;CURRENT PROGRAM START ADDRESS.
CURTST: OPEN ;CONTAINS ADDR OF CURRENT TEST.
RTNNO: OPEN ;CONTAINS CURRENT TEST #.
NXTST: OPEN ;CONTAINS ADDR OF NEXT TEST.
ICTR: OPEN ;CONTAINS CURRENT ITERATION COUNT
SCOPTR: OPEN ;CONTAINS CURRENT SCOPE POINTER.
FRGID: OPEN ;CONTAINS PROGRAM INDICATORS
PRGTAB: PRG0 ;PRG0 START ADDRESS
        PRG1 ;PRG1 START ADDRESS
        PRG2 ;PRG2 START ADDRESS
        PRG3 ;PRG3 START ADDRESS
        PRG4 ;PRG4 START ADDRESS
        PRG5 ;PRG5 START ADDRESS
        PRG6 ;PRG6 START ADDRESS
        PRG7 ;PRG7 START ADDRESS
        PRG10 ;PRG10 START ADDRESS
        PRG11 ;PRG11 START ADDRESS

```


158
159 001200 003764
160 001202 004310
161 001204 004406
162 001206 001530
163 001210 001432
164 001212 001274
165 001214 003614
166 001216 003644
167 001220 001316
168 001222 003674
169 001224 002310
170 001226 003034
171 001230 003050
172 001232 003046
173 001234 005256
174 001236 001520
175 001240 003410
176 001242 003450
177 001244 003066
178 001246 004342
179 001250 000000
180 001252 000000
181 001254 000000
182 001256 000000
183 001260 000000
184 001262 000000
185 001264 000000
186 001266 000000
187 001270 000000
188 001272 000000

EMTTAB: TYP ; POINTER TO TYPEOUT ROUTINE
TYP5 ; POINTER TO CHAINED MESSAGES ROUTINE
STAL ; POINTER TO RANDOM STALL ROUTINE
ERR ; POINTER TO ERROR ROUTINE
DTCHK
CHLT ; COMMON HALT
STLSRV
STLSPV
EHLT ; POINTER TO ERROR HALT ROUTINE.
SRSETT
CHAINN
CHK33
CHK35
CHK330
TYPL3
DTHLT
SAVRG
RSTRG
CKASR

TRPTAB: DLY
RCNT: OPEN ; CHARACTER COUNT
CRBUF: OPEN ; HOLDS ONE CHARACTER FROM READER.
CHR1: OPEN
CHR2: OPEN
CHR3: OPEN
ERCTR: OPEN
CTRA: OPEN
CTRB: OPEN
CTRC: OPEN
CTRD: OPEN

```

189
190
191 001274 022767 000000 176540 :COMMON HALT ROUTINE
192 001302 001004 CHLT:  CMP #0,MONITR ;MONITOR LOAD?
193 001304 011600      BNE  CHLTR  ;YES - EXIT
194 001306 162700 000002      MOV  @%6,%0 ;DEVELOP ADDRESS OF CALLER
195 001312 000000      SUB  #2,%0
196 001314 000002      HALT ;HALT, ADDRESS OF CALL INSTRUCTION
197      CHLTR: RTI ;IN DATA LIGHTS
198 001316 022767 000000 176516 :UNCONDITIONAL ERROR HALT ROUTINE
199 001324 001435 EHLT:  CMP #0,MONITR ;MONITOR LOAD?
200 001326 104011      BEQ  EHLTA  ;NO - HLT.
201 001330 004767 001556      SRESET
202 001334 016700 002040      JSR  %7,CONIT ;SET UP CONSOLE POINTERS
203 001340 162700 000010      MOV  CURADD,%0 ;SET UP DEVICE NUMBER
204 001344 012767 016244 003346      SUB  #10,%0 ;OF DISPLAY IN ERROR.
205 001352 004767 000330      MOV  #ERRORA,A1ST ;SET ERROR DEVICE ADDRESS
206 001356 112777 000067 003334      JSR  %7,ACNVE ;CONVERT OCTAL TO ASCII
207 001364 011600      MOVB #67,@A1ST
208 001366 162700 000002      MOV  @%6,%0 ;TRANSFER PROGRAM ERROR
209 001372 012767 016273 003320      SUB  #2,%0 ;ADDRESS TO MESSAGE
210 001400 004767 000302      MOV  #ERRORB,A1ST
211 001404 104000      JSR  %7,ACNVE
212 001406 016202      TYPE ;TYPE ERROR MESSAGE
213 001410 004767 001554      ERRORM
214 001414 000167 000452      JSR  %7,RSTART
215 001420 011600      JMP  NSTART
216 001422 162700 000002      EHLTA: MOV @%6,%0 ;DEVELOP ADDRESS OF CALLER
217 001426 000000      SUB  #2,%0
218 001430 000002      ERRH:  HALT ;HALT ADDRESS OF ERROR CALL
219      RTI ;IN DATA LIGHTS.
220 001432 126767 177614 177613 :DATA CHECK ROUTINE.
221 001440 001432 DTCHK: CMPB CRBUF,CRBUF+1 ;COMPARE EXPECTED AND RECEIVED
222 001442 022767 000000 176372      BEQ  DTCHKA ;CHARS. BRANCH IF SAME.
223 001450 001423      CMP  #0,MONITR ;MONITOR LOAD?
224 001452 004767 001434      BEQ  DTHLT ;NO GO TO NORMAL HALT
225 001456 016700 001716      JSR  %7,CONIT ;SET UP CONSOLE POINTERS
226 001462 162700 000010      MOV  CURADD,%0 ;SET UP DEVICE NUMBER OF
227 001466 012767 016406 003224      SUB  #10,%0 ;DISPLAY IN ERROR
228 001474 004767 000206      MOV  #DERRA,A1ST ;SET ERROR MESSAGE ADDRESS
229 001500 112777 000067 003212      JSR  %7,ACNVE ;CONVERT OCTAL TO ASCII
230 001506 104000      MOVB #67,@A1ST ;MOST SIG.DIGIT = 7
231 001510 016361      TYPE ;TYPE ERROR MESSAGE
232 001512 004767 001452      DERR
233 001516 000565      JSR  %7,RSTART
234 001520 016700 177526      BR  NSTART
235 001524 000000      DTHLT: MOV CRBUF,%0 ;MOVE S/B AND WAS CHARS TO RD.
236      DERRH: HALT ;DATA ERROR HALT. GOOD CHAR IN
237 001526 000002      ;DATA LIGHTS LEFT. BAD CHAR IN DATA
238      DTCHKA: RTI ;LIGHTS RIGHT. EXIT.
239 001530 032767 040000 176032 :CONDITIONAL ERROR HALT.
240 001536 001404      ERR:  BIT #SCOPSW,SR ;CHECK SCOPE SWITCH.
241 001540 005767 177406      BEQ  ERRA ;BRANCH IF NO SCOPE DESIRED.
242 001544 100001      TST  PRGID ;SCOPING WANTED. FIRST ERROR?
243 001546 000002      BPL  ERRA ;NO SCOPE IF FIRST ERROR.
244 001550 052767 100000 177374      RTI ;SCOPE EXIT.
245      ERRH: BIS #BIT15,PRGID ;SET ERROR INDICATOR.

```

245	001556	022767	000000	176256	CMP	#0,MUNITR	;MONITOR LOAD?
246	001564	001434			BEQ	ERRB	;NO - GO TO NORMAL HALT
247	001566	104011			SRESET		
248	001570	004767	001316		JSR	%7,CONIT	;SET UP CONSOLE POINTERS
249	001574	016700	001600		MOV	CURADD,%0	;SET UP DEVICE NUMBER
250	001600	162700	000010		SUB	#10,%0	;OF DISPLAY IN ERROR
251	001604	012767	016323	003106	MOV	#CERRA,A1ST	;SET ERROR MESSAGE ADDRESS
252	001612	004767	000070		JSR	%7,ACNVE	;CONVERT OCTAL TO ASCII
253	001616	112777	000067	003074	MOVB	#6,A1ST	;MOST SIG DIGIT = 7
254	001624	012767	016352	003066	MOV	#CERRB,A1ST	;TRANSFER PROGRAM ERROR
255	001632	011600			MOV	%6,%0	;ADDRESS TO MESSAGE
256	001634	162700	000002		SUB	#2,%0	
257	001640	004767	000042		JSR	%7,ACNVE	;CONVERT OCTAL TO ASCII
258	001644	104000			TYPE		;TYPE ERROR MESSAGE
259	001646	016303			CERR		
260	001650	004767	001314		JSR	%7,RSTART	
261	001654	000506			BR	NSTART	
262	001656	011600			ERRB: MOV	%6,%0	;DEVELOP ADDRESS OF
263	001660	162700	000002		SUB	#2,%0	;CALLER
264	001664	000000			CERRH: HALT		;ERROR HALT
265	001666	000002			RTI		;EXIT
266							;ROUTINE END HALT SUBROUTINE.
267	001670	005767	175674		SHALT: TST	SR	;CHECK HALT SWITCH.
268	001674	100003			BPL	SHLTA	;BRANCH IF NO HALT DESIRED.
269	001676	116700	177240		MOVB	RTNNO,%0	;CURRENT TEST # TO RD.
270	001702	000000			RHLTA: HALT		;ROUTINE END HALT.
271	001704	000207			SHLTA: RTS	%7	;EXIT.
272	001706	016701	003006		ACNVE: MOV	A1ST,%1	;TRANSFER ADDRESS TO R1
273	001712	062701	000006		ADD	#6,%1	;ADD 6 TO REF LSD
274	001716	012702	000006		MOV	#6,%2	;SET ROUTINE CTR TO 6
275	001722	010067	003000		MOV	%0,ACNVX	;TRANSFER NUMBER TO CONVERT ROUTINE REG
276	001726	000167	003012		JMP	ACNVM	;GO TO CONVERT ROUTINE

```

277
278          :ROUTINE FOR MONITOR LOAD
279 001732 012706 001104          START:  MOV    #SPBOT,%6      ;SET BOTTOM OF STACK
280 001736 012767 002724 176060  MOV    #PFAIL,29    ;SET POWER FAIL POINTER
281 001744 016767 001432 001426  MOV    LOWADD,CURADD ;INITIALIZE POINTER FOR START OF MULTIPLES
282 001752 012767 002010 176024  MOV    #END,4       ;INITIALIZE TIME OUT TRAP
283 001760 012767 000300 001420  MOV    #300,CURVEC  ;INITIALIZE VECTOR POINTER
284 001766 012700 174000          MOV    #174000,%0   ;ADDRESS OF FIRST DC11 TO RO
285 001772 005710          B:      TST    %0     ;IS THIS DEVICE EQUIPPED?
286 001774 062767 000010 001404  ADD    #10,CURVEC   ;NO ADD #10 TO VECTOR AND
287 002002 062700 000010          ADD    #10,%0      ;ADDRESS
288 002006 000771          BR      B
289 002010 022626          END:    POPSP2
290 002012 016702 001364          MOV    LOWADD,%2
291 002016 162702 176500          SUB    #176500,%2  ;SUBTRACT GROUP OFFSET
292 002022 060267 001360          ADD    %2,CURVEC  ;ADD GROUP BIAS TO VECTOR POINTER.
293 002026 004767 001060          JSR    %7,CONIT   ;RESTORE POINTERS & VECTORS TO CONSOLE
294 002032 012767 000000 001336  MOV    #0,STNUM    ;RESET PROGRAM LOADER
295 002040 022767 000000 175774  CMP    #0,MONITR  ;IS THIS A MONITOR LOAD?
296 002046 001427          BEQ    STARTO     ;NO - GO TO REGULAR START
297 002050 022767 000000 175512  CMP    #0,SR      ;ANY PANEL SW SET?
298 002056 001405          BEQ    NSTART     ;NO - GO TO NEW START
299 002060 104000          TYPE   ;TYPE ABORT MESSAGE
300 002062 016415          ABORT
301 002064 000005          RESET
302 002066 000167 011214          JMP    LOGICAL    ;RETURN TO MONITOR.
303 002072 012706 001104          NSTART: MOV    #SPBOT,%6 ;SET BOTTOM OF SP STACK
304 002076 005067 175674          CLR    PSW
305 002102 012767 000006 175674  MOV    #6,MACHER
306 002110 005067 177026          CLR    RTNNO
307 002114 016700 001256          MOV    STNUM,%0   ;TRANSFER PROGRAM NUMBER TO RO
308 002120 005267 001252          INC    STNUM
309 002124 000422          BR     CRTA
310 002126 012706 001104          STARTO: MOV   #SPBOT,%6 ;SET BOTTOM OF SP STACK.
311 002132 005067 175640          CLR    PSW
312 002136 012767 000006 175640  MOV    #6,MACHER
313 002144 005067 176772          CLR    RTNNO
314 002150 016700 175414          MOV    SR,%0
315 002154 042700 177760          BIC    #177760,%0 ;(SR) TO RO
316 002160 020027 000014          CMP    %0,#14    ;LIMIT (SR) TO BITS 3-0
317 002164 101402          BLOS  CRTA       ;COMPARE (SR) TO PROGRAM LIMIT
318 002166 104010          INCPRG: EHALT   ;VALID PROGRAM NUMBER?
319 002170 000660          BR     START     ;NO. INCORRECT PRG NUMBER
320 002172 005067 176754          CRTA:  CLR     ;START OVER.
321 002176 010067 176732          MOV    PRGID
322 002202 006100          ROL    %0,PRGNUM ;SAVE PROGRAM NUMBER AT PRGNUM
323 002204 000170 001154          JMP    %PRGTAB(0) ;ROX2
324 002210 104005          SRSET: CHALT    ;GO TO SELECTED PROGRAM.
325 002212 016767 176720 176724  GETRDY: MOV    KSTART,NXTST ;SET SR OPTIONS DESIRED
326 002220 000167 000350          JMP    CLEAN     ;ADDR OF 1ST ROUTINE TO NXTST
327 002224 004767 000240          GTRDYA: JSR    %7,FORWD ;GO CLEAN UP.
328 002230 032767 001000 175332  GTRDYB: BIT    #SRTSW,SR ;ROLL FORWARD TO "NEXT" ROUTINE.
329 002236 001003          BNE   GTRDYC    ;CHECK FOR SELECT ROUTINE SWITCH
330 002240 004767 000302          JSR    %7,GOTST ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
331 002244 000460          BR     CHNB     ;GO RUN CURRENT ROUTINE.
332 002246 016700 175316          GTRDYC: MOV   SR,%0 ;NO GO. MANUAL RTN BYPASSED.
          ;(SR) TO RO

```

333	002252	042700	177600			BIC	#177600,%0	:MASK UNDESIRED BITS
334	002256	126700	176660			CMPB	RTNNO,%0	:COMPARE RTNNO TO (R0)
335	002262	001004				BNE	GTRDY0	:BRANCH IF ROUTINE NOT FOUND YET.
336	002264	004767	000256			JSR	%7,GOTST	:GO RUN ROUTINE
337	002270	104010				EHALT		:NO GO. MANUAL RTN SELECTED BYPASSED.
338	002272	000747				BR	GETRDY	
339	002274	022767	177777	176642	GTRDYD:	CMP	#-1,NXTST	:NO. CHECK FOR LAST ROUTINE.
340	002302	001350				BNE	GTRDYA	:LAST ROUTINE?
341	002304	104010			INCRTN:	EHALT		:YES. INCORRECT ROUTINE SELECTED.
342	002306	000741				BR	GETRDY	:START OVER.
343	002310	005767	176636		CHAINN:	TST	PRGID	:TEST ERROR BIT IN PRGID.
344	002314	100013				BPL	CHNA	:BRANCH IF ERROR BIT NOT SET.
345	002316	032767	040000	175244		BIT	#SCOPSW,SR	:ERROR BIT SET. CHECK FOR SCOPE OPTION.
346	002324	001407				BEQ	CHNA	:SCOPE SWITCH SET IN SR?
347	002326	022767	177777	176614		CMP	#-1,SCOPTR	:YES. CHECK SCOPE ENTRY POINTER
348	002334	001403				BEQ	CHNA	:BRANCH IF SCOPE ENTRY IS -1.
349	002336	017716	176606			MOV	@SCOPTR,@%6	:SET UP TO GO SCOPING
350	002342	000002				RTI		:GO TO SCOPE ENTRY.
351	002344	042767	100000	176600	CHNA:	BIC	#BIT15,PRGID	:CLEAR ERROR BIT IN PRGID.
352	002352	032767	004000	175210		BIT	#NITRSW,SR	:TEST INHIBIT ITERATION SWITCH
353	002360	001004				BNE	CHNAA	:INHIBIT ITERATION?
354	002362	005367	176560			DEC	ICTR	:NO
355	002366	001401				BEQ	CHNAA	:COUNT 0?
356	002370	000002				RTI		:NO. RETURN TO TEST ROUTINE
357	002372	022626			CHNAA:	POPSP2		:POP STACK TWICE
358	002374	104011				SRESET		
359	002376	004767	001106			JSR	%7,RETPVT	:RESTORE VECTOR TABLES
360	002402	004767	177262			JSR	%7,SHALT	:GO HALT IF HALT SWITCH IS SET

361											
362	002406	032767	001000	175154	CHNB:	BIT	#SRTSW,SR				;CHECK SELECT ROUTINE SWITCH
363	002414	001276				BNE	GETRDY				;SELECT ROUTINE SWITCH SET?
364	002416	022767	177777	176520		CMP	#-1,NXTST				;NO.
365	002424	001275				BNE	GTRDYA-4				;LAST TEST?
366	002426	032767	002000	175134		BIT	#LPRGSW,SR				;YES. TEST LOOP PROGRAM SWITCH.
367	002434	001266				BNE	GETRDY				;LOOP PROGRAM?
368	002436	022767	000000	175376		CMP	#0,MONITR				;MONITOR LOAD?
369	002444	001407				BEQ	PRGEND				;YES - GO TO HALT
370	002446	026727	000724	000001		CMP	STNUM,#1				;PROGRAM #1 COMPLETED?
371	002454	003606				BLE	NSTART				;YES GO TO NEW START
372	002456	004767	000506			JSR	%7,RSTART				;GO TO RESTART
373	002462	000603				BR	NSTART				
374	002464	000000			PRGEND:	HALT					;NO. PROGRAM END.
375	002466	000747				BR	CHNB				
376	002470	016705	176450		FORWD:	MOV	NXTST,%5				;ADDR OF NEXT ROUTINE TO R5.
377	002474	012567	176442			MOV	(5)+,RTNNO				;GET NEXT ROUTINE NUMBER.
378	002500	012567	176440			MOV	(5)+,NXTST				;GET ADDR OF NEXT "NEXT" ROUTINE.
379	002504	105767	176442			MOV	PRGIO				;CHECK IF PROGRAM SCOPE AND I COUNT
380	002510	100407				TSTB	FORWDB				;PARAMETERS. BRANCH IF NOT.
381	002512	012567	176430			BMI	(5)+,ICTR				;GET ITERATION COUNT.
382	002516	012567	176426			MOV	(5)+,SCOPTR				;GET SCOPE LOOP ENTRY POINTER.
383	002522	010567	176412		FORWDA:	MOV	%5,CURTST				;ADDR OF NOW CURRENT TEST TO CURTST.
384	002526	000207				RTS	%7				;EXIT FORWD SUBROUTINE.
385	002530	012767	177777	176412	FORWDB:	MOV	#-1,SCOPTR				;FORCE "NO SCOPE"
386	002536	012767	000001	176402		MOV	#1,ICTR				;FORCE I COUNT OF 1
387	002544	000766				BR	FORWDA				
388	002546	005767	176370		GOTST:	TST	RTNNO				;CHECK FOR MANUAL RTN.
389	002552	100005				BPL	GOTSTA				;BRANCH IF NOT MANUAL RTN.
390	002554	032767	000400	175006		BIT	#BYPMAN,SR				;MANUAL RTN. BYPASS IT?
391	002562	001401				BEQ	GOTSTA				;NO. RUN IT.
392	002564	000207				RTS	%7				;BYPASS MANUAL ROUTINE.
393	002566	005726			GOTSTA:	POPSP					
394	002570	000177	176344			JMP	@CURTST				;GO RUN TEST.
395	002574	012767	001233	001156	CLEAN:	MOV	#1233,RP1				;RESET RANDOM NOS.
396	002602	012767	005622	001152		MOV	#5622,RP2				
397	002610	104011				SRESET					
398	002612	012767	000006	175164		MOV	#6,MACHER				;RESET MACHER TRAP.
399	002620	005067	175152			CLR	PSW				
400	002624	012706	001104			MOV	#SPBOT,%6				;SET UP BOTTOM OF STACK.
401	002630	000167	177370			JMP	GTRDYA				
402	002634	011646			EMTINT:	MOV	@%6,-(6)				;GET SAVED PC.
403	002636	162716	000002			SUB	#2,@%6				;DECREMENT PC BY 2.
404	002642	017616	000000			MOV	@(6),@%6				
405	002646	121627	000022			CMPB	@%6,#22				;CHECK THAT CALL IS
406	002652	101402				BLOS	EMTA				;WITHIN LIMITS.
407	002654	000000				HALT					;CALL NOT WITHIN LIMITS.
408	002656	000776				BR	.-2				
409	002660	006116			EMTA:	ROL	@%6				;EMT ARG X 2.
410	002662	042716	177001			BIC	#177001,@%6				;REMOVE 7 MSB.
411	002666	062716	001200			ADD	#EMTTAB,@%6				;FORM EMT RTN ADDR.
412	002672	017616	000000			MOV	@(6),@%6				
413	002676	000136				JMP	@(6)+				;GO TO EMT ROUTINE.
414	002700	011646			TRPINT:	MOV	@%6,-(6)				;GET SAVED PC.
415	002702	162716	000002			SUB	#2,@%6				;DECREMENT PC BY 2.
416	002706	017616	000000			MOV	@(6),@%6				

M03

KL11 TESTS FOR VT05 DISPLAY TERMINAL
DZVTB.P11

MACY11 27(732) 20-SEP-76 10:11 PAGE 10

SEQ 0038

417	002712	121627	000000		CMPB	%6,#0		;CHECK THAT EMT
418	002716	101436			BLOS	TRPA		;IS WITHIN LIMITS.
419	002720	000000			HALT			;TRAP CALL NOT IN LIMIT.
420	002722	000776			BR			
421						.-2		
422	002724	010046			:ENTER HERE FOR POWER FAIL			
423	002726	010146			PFail: MOV	%0,-(6)		;SAVE REGISTER OR STACK
424	002730	010246			MOV	%1,-(6)		;WHEN POWERING DOWN
425	002732	010346			MOV	%2,-(6)		
426	002734	010446			MOV	%3,-(6)		
427	002736	010546			MOV	%4,-(6)		
428	002740	016746	175060		MOV	%5,-(6)		
429	002744	010667	000010		MOV	24,-(6)		
430	002750	012767	002762	175046	MOV	%6,SAVR6		;STORE STACK POSITION
431	002756	000000			MOV	*RESTART,24		
432	002760	000000			HALT			;HALT ON POWER DOWN NORMAL
433	002762	016706	177772		SAVR6: 0			;STACK IS SAVED HERE
434	002766	012667	175032		RESTART: MOV	SAVR6,%6		;RESTORE REGISTER OFF STACK
435	002772	012605			MOV	(6)+,24		;WHEN POWERING UP
436	002774	012604			MOV	(6)+,%5		
437	002776	012603			MOV	(6)+,%4		
438	003000	012602			MOV	(6)+,%3		
439	003002	012601			MOV	(6)+,%2		
440	003004	012600			MOV	(6)+,%1		
441	003006	000167	176720		MOV	(6)+,%0		
442	003012	000002			JMP	START		
					RTI			;RETURN TO MAIN LINE

443									
444	003014	006116			TRPA:	ROL	2%6		;TRAP ARG X 2.
445	003016	042716	177001			BIC	#177001,2%6		;REMOVE 7 MSB.
446	003022	062716	001246			ADD	#TRPTAB,2%6		;FORM TRAP RTN ADDR.
447	003026	017616	000000			MOV	2(6),2%6		
448	003032	000136				JMP	2(6)+		;GO TO TRAP ROUTINE.
449	003034	005767	176072		CHK33:	TST	TTYTYP		;CHECK FOR 33.
450	003040	001002				BNE	.+6		;BRANCH IF NOT 33.
451	003042	062716	000002			ADD	#2,2%6		;+2 TO EXIT POINTER
452	003046	000002			CHK330:	RTI			;EXIT
453	003050	022767	000001	176054	CHK35:	CMP	#1,TTYTYP		;CHECK FOR 35.
454	003056	001002				BNE	.+6		;BRANCH IF NOT 35.
455	003060	062716	000002			ADD	#2,2%6		;+2 TO EXIT POINTER
456	003064	000002				RTI			;EXIT
457	003066	032767	000010	176036	CKASR:	BIT	#BIT3,TTYTYP		;CHECK FOR ASR TTY.
458	003074	001001				BNE	.+4		;BRANCH IF NOT ASR.
459	003076	000002				RTI			;ASR. EXIT.
460	003100	022626				POPSP2			;POP STACK TWICE.
461	003102	012767	000001	176036		MOV	#1,ICTR		;FORCE I COUNT TO A 1.
462	003110	104012				CHAIN			;CHAIN TO BYPASS ROUTINE.
463						;SUBROUTINE TO RESTORE POINTERS AND VECTORS FOR CONSOLE OUTPUT.			
464	003112	016700	175770		CONIT:	MOV	CONADD,%0		;CONSOLE ADDRESS TO RO
465	003116	010067	175770			MOV	%0,TKS		;SET LSR CSR TO CONSOLE
466	003122	005720				TST	(0)+		;ADD 2 TO RO
467	003124	010067	175764			MOV	%0,TKB		;SET LSR BUFFER TO CONSOLE
468	003130	005720				TST	(0)+		;ADD 2 TO RO
469	003132	010067	175760			MOV	%0,TPS		;SET LSP CSR TO CONSOLE
470	003136	005720				TST	(0)+		;ADD 2 TO RO
471	003140	010067	175754			MOV	%0,TPB		;SET LSP BUFFER TO CONSOLE
472	003144	016767	175740	175750		MOV	CONVEC,TKVTR		;SET LSR INTERRUPT VECTOR TO CONSOLE
473	003152	016767	175732	175746		MOV	CONVEC,TPVTR		;SET LSP INTERRUPT VECTOR TO CONSOLE
474	003160	062767	000004	175740		ADD	#4,TPVTR		
475	003166	000207				RTS	%7		
476						;SUBROUTINE FOR ADVANCING TEST.			
477	003170	012767	000000	000200	RSTART:	MOV	#0,STNUM		;RESET PROGRAM LOADER
478	003176	022767	000000	174364		CMP	#0,SR		;ANY PR SWITCH SET?
479	003204	001407				BEQ	RSTA		;NO - CONTINUE
480	003206	004767	177700			JSR	%7,CONIT		;RESTORE POINTER TO CONSOLE
481	003212	104000				TYPE			;TYPE ABORT MESSAGE
482	003214	016415				ABORT			
483	003216	000005				RESET			
484	003220	000167	010062			JMP	LOGICAL		;EXIT TO MONITOR
485	003224	004767	000260		RSTA:	JSR	%7,RETPVT		;RESET INTERRUPT VECTORS
486	003230	016700	000144			MOV	CURADD,%0		;NEXT DEVICE ADD TO RO
487	003234	026767	000140	000142		CMP	CURADD,LSTADD		;IS THIS LAST ADDRESS
488	003242	001442				BEQ	RSTB		;YES - GO TO EXIT
489	003244	062767	000010	000126		ADD	#10,CURADD		;COMPUTE ADDRESS FOR NEXT TEST.
490	003252	010067	175634			MOV	%0,TKS		;NEW LSR CSR
491	003256	062700	000002			ADD	#2,%0		
492	003262	010067	175626			MOV	%0,TKB		;NEW LSR BUFFER
493	003266	062700	000002			ADD	#2,%0		
494	003272	010067	175620			MOV	%0,TPS		;NEW LSP CSR
495	003276	062700	000002			ADD	#2,%0		
496	003302	010067	175612			MOV	%0,TPB		;NEW LSP BUFFER
497	003306	016700	000074			MOV	CURVEC,%0		;NEXT VECTOR ADD TO RO
498	003312	062767	000010	000066		ADD	#10,CURVEC		;COMPUTE ADDRESS FOR NEXT VECTOR


```

499 003320 010067 175576      MOV      %0,TKVTR      ;NEW LSR INTERRUPT VECTOR
000 003324 062700 000004      ADD      #4,%0
001 003330 010067 175572      MOV      %0,TPVTR      ;NEW LSP INTERRUPT VECTOR
002 003334 012767 003350 174442      MOV      #RSTB,4      ;SETUP TRAP EXIT
003 003342 005777 175544      TST      %TKS          ;IS THIS DEVICE EQUIPPED?
004 003346 000412          BR       RSTC          ;EXIT
005 003350 016767 000024 000026 RSTB:  MOV      CURADD,LSTADD ;SETUP LAST ADDRESS
006 003356 004767 177530      JSR      %7,CONIT      ;RESTORE POINTER TO CONSOLE
007 003362 104000          TYPE          ;TYPE DONE MESSAGE
008 003364 016160          DONE
009 003366 000005          RESET
010 003370 000167 007712      JMP      LOGICAL      ;EXIT TO MONITOR
011 003374 000207          RSTC:  RTS          %7
012 003376 000000          STNUM:  OPEN
013 003400 000000          CURADD: OPEN
014 003402 176500          LOWADD: 176500
015 003404 000000          LSTADD: OPEN
016 003406 000000          CURVEC: OPEN
017 003410 012667 000030      SAVRG:  MOV      (6)+,SVRPC      ;SAVE PC AND PSW.
018 003414 012667 000026      MOV      (6)+,SVRPSW
019 003420 010446          MOV      %4,-(6)      ;SAVE REGS 0 - 4
020 003422 010346          MOV      %3,-(6)      ;IN STACK.
021 003424 010246          MOV      %2,-(6)
022 003426 010146          MOV      %1,-(6)
023 003430 010046          MOV      %0,-(6)
024 003432 016746 000010      MOV      SVRPSW,-(6)   ;RESTORE PC AND PSW.
025 003436 016746 000002      MOV      SVRPC,-(6)
026 003442 000002          RTI          ;EXIT.
027 003444 000000          SVRPC:  OPEN
028 003446 000000          SVRPSW: OPEN
029 003450 012667 000030      RSTRG:  ;RESTORE REGS 0 TO 4 SUBROUTINE.
030 003454 012667 000025      MOV      (6)+,RSTPC      ;SAVE PC AND PSW.
031 003460 012600          MOV      (6)+,RSTPSW
032 003462 012601          MOV      (6)+,%0      ;RESTORE REGS 0 - 4
033 003464 012602          MOV      (6)+,%1      ;FROM STACK.
034 003466 012603          MOV      (6)+,%2
035 003470 012604          MOV      (6)+,%3
036 003472 016746 000010      MOV      RSTPSW,-(6)   ;RESTORE PC AND PSW.
037 003476 016746 000002      MOV      RSTPC,-(6)
038 003502 000002          RTI          ;EXIT
039 003504 000000          RSTPC:  OPEN
040 003506 000000          RSTPSW: OPEN
041 003510 016701 175406      ;SUBROUTINE TO RESET INTERRUPT VECTORS
042 003514 010100      RETPVT: MOV      TKVTR,%1      ;PLACE CURRENT TRAP VECTOR
043 003516 062700 000002      MOV      %1,%0          ;ADDRESS IN R1, AND R1 TO R0.
044 003522 010021          ADD      #2,%0          ;ADD #2 TO R0
045 003524 012721 000000      MOV      %0,(1)+      ;STORE .+2 AT TKVTR
046 003530 062700 000004      MOV      #0,(1)+      ;STORE HALT AT TKVTR+2
047 003534 010021          ADD      #4,%0
048 003536 012711 000000      MOV      %0,(1)+      ;STORE .+2 AT TPVTR
049 003542 000207          MOV      #0,%1          ;STORE HALT AT TPVTR+2
050          RTS          %7

```

```

553
554
555
556 003544 012767 000310 000040
557 003552 052777 000004 175335
558 003560 005077 175334
559 003564 105777 175322
560 003570 100407
561 003572 104400
562 003574 000001
563 003576 005367 000010
564 003602 001370
565 003604 104010
566 003606 000756
567 003610 000207
568 003612 000000
569
570 003614 017667 000000 000012
571 003622 062716 000002
572 003626 016701 175270
573 003632 012721 000000
574 003636 016721 175262
575 003642 000002
576
577 003644 017667 000000 000012
578 003652 062716 000002
579 003656 016701 175244
580 003662 012721 000000
581 003666 016721 175236
582 003672 000002
583 003674 012700 052525
584 003700 005100
585 003702 010067 177770
586 003706 000005
587 003710 000002
588
589 003712 016700 000042
590 003716 006100
591 003720 006100
592 003722 066700 000034
593 003726 010067 000026
594 003732 006100
595 003734 006100
596 003736 066700 000020
597 003742 006100
598 003744 006100
599 003746 010067 000010
600 003752 016700 000002
601 003756 000207
602 003760 001233
603 003762 005622

```

```

:ROUTINE TO FETCH A CHARACTER
AREAD: MOV #200, BRCTR ;SET UP DELAY COUNT.
      BIS #4, JTP5 ;SET MAINTENANCE BIT
      CLR JTP8 ;LOAD PUNCH BUFFER
ARDA: TSTB JTK5 ;CHECK DONE BIT.
      SMI ARDB ;BRANCH IF DONE.
      DELAY ;DELAY 1 MILLISECOND.
      DEC BRCTR ;TIME UP?
      BNE ARDA ;BRANCH IF TIME NOT UP YET.
      EHALL ;ERROR. NO RESPONSE FROM READER.
      BR AREAD ;TRY AGAIN.
ARDB: RTS ;EXIT
BRCTR: OPEN

:ROUTINE TO SET LSR INTERRUPT VECTOR AND PRIORITY
STLSRV: MOV 2(6), STPRA+2 ;MOVE VECTOR ADDR TO STPRA+2
        ADD #2, 2(6) ;SET UP EXIT
        MOV TKVTR, %1
STPRA: MOV #OPEN, (1)+ ;SET VECTOR ADDRESS
        MOV TKLVL, (1)+ ;SET PRIORITY
        RTI ;EXIT

:ROUTINE TO SET LSP INTERRUPT VECTOR AND PRIORITY.
STLSPV: MOV 2(6), STPPA+2 ;MOVE VECTOR ADDR TO STPPA+2
        ADD #2, 2(6) ;SET UP EXIT
        MOV TPVTR, %1
STPPA: MOV #OPEN, (1)+ ;SET VECTOR ADDRESS.
        MOV TPLVL, (1)+ ;SET PRIORITY
        RTI ;EXIT.

:ROUTINE TO ISSUE RESET.
SRSETT: MOV #52525, %0 ;DATA TO R0.
        CCM %0 ;COMPLEMENT (R0).
        MOV %0, SRSETT+2 ;(R0) TO SRSETT+2.
        RESET ;ISSUE RESET. (R0) IS
        RTI ;DISPLAYED. EXIT.

:RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER C.
RNGEN: MOV RP1, %0
        ROL %0
        ROL %0
        ADD RP2, %0
        MOV %0, RP1
        ROL %0
        ROL %0
        ADD RP2, %0
        ROL %0
        ROL %0
        MOV %0, RP2
        MOV RP1, %0
        RTS ;EXIT. NUMBER IN R0
        %7

```

```

RP1: 1233
RP2: 5622

```

```

604 ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
605 003764 011600 TYP: MOV #0,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
606 003766 062716 000002 ADD #2,%6 ;SET UP EXIT.
607 003772 011000 MOV #0,%0 ;ADDRESS OF MESSAGE TO RO.
608 003774 112067 000306 TYP A: MOVB (0)+,TYPDAT ;GET CHARACTER
609 004000 122767 000100 000300 CMPB #100,TYPDAT ;CHECK FOR"@"CHARACTER
610 004006 001013 SNE TYP ;BRANCH IF NOT"@".
611 004010 112767 000000 000270 MOVB #0,TYPDAT ;NULL TO CLEAR DOUBLE BUFFER
612 004016 004767 000046 JSR %7,OUTTYP
613 004022 112767 000000 000256 MOVB #0,TYPDAT
614 004030 004767 000034 JSR %7,OUTTYP
615 004034 000002 RTI ;TERMINATOR CHAR. DONE. EXIT.
616 004036 122767 000045 000242 TYP C: CMPB #45,TYPDAT ;CHECK FOR"%".
617 004044 001505 BEQ TYP ;BRANCH IF"%".
618 004046 122767 000043 000232 CMPB #43,TYPDAT ;NOT"%".CHECK FOR".".
619 004054 001506 BEQ TYP ;BRANCH IF ".".
620 004056 004767 000006 JSR %7,OUTTYP ;TYPE CHAR IN TYPDAT
621 004062 000744 BR TYP A
622 ;SUBROUTINE TO INSERT NULLS AND PRINT CHARACTER. USES REGISTER
623 ;LOCATION TYPDAT REFERENCED IN THE OUTPUT MESSAGE SUBROUTINE
624 ;AND LINKS TO TYPDAT REGISTER IN THE LSPCH SUBROUTINE.
625 004064 110067 000216 OUT: MOVB %0,TYPDAT ;MOVE LSPCH CHAR TO TYPDAT.
626 004070 122767 000012 000210 OUTTYP: CMPB #12,TYPDAT ;CHECK FOR LINE FEED.
627 004076 001437 BEQ OUTNUL ;GO TO NULL ROUTINE
628 004100 122767 000013 000200 CMPB #13,TYPDAT ;CHECK FOR CURSOR DWN
629 004106 001433 BEQ OUTNUL ;GO TO NULL ROUTINE
630 004110 122767 000035 000170 CMPB #35,TYPDAT ;CHECK FOR CURSOR HOME
631 004116 001427 BEQ OUTNUL ;GO TO NULL ROUTINE
632 004120 122767 000032 000160 CMPB #32,TYPDAT ;CHECK FOR CURSOR UP
633 004126 001423 BEQ OUTNUL ;GO TO NULL ROUTINE
634 004130 122767 000037 000150 CMPB #37,TYPDAT ;CHECK FOR ERASE SCREEN
635 004136 001417 BEQ OUTNUL ;GO TO NULL ROUTINE
636 004140 122767 000016 000140 CMPB #16,TYPDAT ;CHECK FOR CAD OPERATION
637 004146 001407 BEQ OUTNCT ;GO TO CAD NULL ROUTINE
638 004150 026727 000102 000004 CMP NULL,#4 ;NULL CTR 4 OR GREATER?
639 004156 003412 BLE OUTCAR ;IF NOT GREATER GO TO OUTPLT
640 004160 005367 000072 DEC NULL ;DEC NULL REGISTER
641 004164 000407 BR OUTCAR ;GO TO OUTPUT.
642 004166 012767 000005 000062 OUTNCT: MOV #5,NULL ;SET NULL CTR TO 6
643 004174 000403 BR OUTCAR ;GO TO OUTPUT
644 004176 012767 000004 000052 OUTNUL: MOV #4,NULL ;SET NULL CTR TO 4.
645 004204 116777 000076 174706 OUTCAR: MOVB TYPDAT,%TPB ;OUTPUT CHARACTER TO PRINTER
646 004212 105777 174700 TSTB %TPB ;WAIT FOR DONE FLAG.
647 004216 100375 BPL .-4
648 004220 026727 000032 000004 CMP NULL,#4 ;IS COUNTER > 4
649 004226 003012 BGT OUTEX ;IF GREATER EXIT
650 004230 022767 000000 000020 CMP #0,NULL ;IS NULL CTR = 0
651 004236 001406 BEQ OUTEX ;IF ZERO EXIT
652 004240 005367 000012 DEC NULL ;DEC NULL CTR
653 004244 112767 000000 000034 MOVB #0,TYPDAT ;TRANSFER ZERO TO OUTPUT
654 004252 000754 BR OUTCAR
655 004254 000207 CUTEX: RTS %7 ;EXIT
656 004256 000000 NULL: OPEN
657 004260 112767 000015 000020 TYP F: MOVB #15,TYPDAT ;MOVE CARRIAGE RETURN CODE TO TYPDAT
658 004266 004767 177576 JSR %7,OUTTYP ;GO TYPE CHAR.
659 004272 112767 000012 000006 TYP G: MOVB #12,TYPDAT ;MOVE LF CODE TO TYPDAT.

```

660	004300	004767	177564		JSR	%7,OUTTYP	;GO TYPE CHAR.
661	004304	000633			BR	TYP A	
662	004306	000000			TYPDAT:	OPEN	
663					;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER		
664	004310	011600			TYP S:	MOV	%6,%0 ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
665	004312	062716	000002			ADD	#2,%6 ;UPDATE TO NEXT MESSAGE ADDRESS
666	004316	011067	000014			MOV	%0,TYP SB ;ADDRESS OF MESSAGE TO TYP SB
667	004322	022767	177777	000006		CMP	#-1,TYP SB ;CHECK FOR TERMINATOR
668	004330	001001				BNE	TYP SA ;BRANCH IF NOT TERMINATOR.
669	004332	000002				RTI	;TERMINATOR, EXIT
670	004334	104000			TYP SA:	TYPE	;CALL ON TYP SUB TO TYPE MESSAGE
671	004336	000000			TYP SB:	OPEN	;ADDRESS OF MESSAGE GOES HERE
672	004340	000763			BR	TYP S	;GO PROCESS NEXT MESSAGE

```

673
674
675 004342 011667 000036
676 004346 062716 000002
677 004352 017746 000026
678 004356 005067 173414
679 004362 012746 002260
680 004366 005316
681 004370 001376
682 004372 005726
683 004374 005316
684 004376 001371
685 004400 005726
686 004402 000002
687 004404 000000
688
689
690 004406 032767 040000 174536
691 004414 001001
692 004416 000002
693 004420 004767 177266
694 004424 046700 000014
695 004430 001404
696 004432 010067 000002
697 004436 104400
698 004440 000000
699 004442 000002
700 004444 000000
701
702 004446 004767 177240
703 004452 046700 000010
704 004456 001773
705 004460 010067 000004
706 004464 000207
707 004466 000000
708 004470 000000

```

```

:SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
DLY:  MOV   2%6,DLCNT      ;GET DELAY COUNT ADDRESS.
      ADD   #2,2%6        ;SET UP EXIT ADDRESS
      MOV   2DLCNT,-(6)   ;DELAY COUNT TO STACK
      CLR   PSW           ;SET PRIORITY 0
DLYA:  MOV   #2260,-(6)   ;1 MSEC COUNT TO STACK
DLYB:  DEC   2%6          ;DECREMENT 1 MSEC COUNT
      BNE   DLYB         ;BRANCH IF NOT 0.
      POPSP              ;ZERO. UNCOVER MSECS. COUNT.
      DEC   2%6          ;DECREMENT IT
      BNE   DLYA         ;BR IF NOT DONE DELAYING
      POPSP              ;DONE
      RTI                ;EXIT.
DLCNT:  OPEN              ;CONTAINS MILLISECONDS COUNT ADDRESS.
:SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL
:DETERMINED BY CONTENTS OF LOC STLMSK.
STAL:  BIT   #BIT14,PRGID ;TEST FOR STALLS ALLOWED.
      BNE   STALAA       ;ALLOWED.
      RTI                ;NOT ALLOWED.
STALAA: JSR   %7,RNGEN    ;GO GET RANDOM NUMBER.
      BIC   STLMSK,%0    ;# IN RO. APPLY STALL MASK.
      BEQ   STALB        ;BRANCH IF RESULT IS 0.
      MOV   %0,STALA
      DELAY              ;DELAY
STALA:  OPEN              ;DELAY COUNT
STALB:  RTI                ;DONE. EXIT.
STLMSK: OPEN              ;STALL MASK.
:SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
GRCNT:  JSR   %7,RNGEN    ;GET RANDOM NUMBER
      BIC   RCMSK,%0     ;APPLY MASK
      BEQ   GRCNT        ;TRY AGAIN IF RESULT 0
      MOV   %0,RNCNT     ;COUNT TO RNCNT
      RTS   %7           ;EXIT.
RCMSK:  OPEN              ;RANDOM CHARACTER MASK.
RNCNT:  OPEN              ;RANDOM CHARACTER COUNT.

```

```

709      :SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
710 004472 012767 177777 000014 INBIN: MOV    *-1,RIND    ;SET ALL VARIABLES
711 004500 004567 000300          JSR    %5,BMOVE    ;TO MINUS 1.
712 004504 004514          RIND
713 004506 004515          RIND+1
714 004510 000013          LI
715 004512 000207          RTS    %7          ;EXIT
716 004514 000000          RIND: OPEN
717 004516 000000          PTO:  OPEN
718 004520 000000          PT1:  OPEN
719 004522 000000          PIND: OPEN
720 004524 000000          PTO:  OPEN
721 004526 000000          PTIP: OPEN

```

```

722
723
724 004530 016767 177762 177762 :SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN RO
725 004536 005167 177756 GTBIN: MOV PTO,PT1 ;PREVIOUS BIN CHAR TO PT1
726 004542 005167 177746 COM PT1
727 004546 001002 COM RIND
728 004550 005267 177744 BNE .+6
729 004554 042767 177400 177736 INC PT1
730 004562 016767 177732 177726 BIC #177400,PT1 ;MASK TO 8 BITS
731 004570 016700 177724 MOV PT1,PTO ;SAVE BIN CHAR IN PTO
732 004574 000207 MOV PT1,%0 ;BIN CHAR TO RO.
733 004576 016767 177722 177722 GTBINP: MOV PTOP,PTIP ;EXIT.
734 004604 005167 177716 COM PTIP ;PREVIOUS BIN CHAR TO PTIP
735 004610 005167 177706 COM PIND
736 004614 001002 BNE .+6
737 004616 005267 177704 INC PTIP
738 004622 042767 177400 177676 BIC #177400,PTIP ;MASK TO 8 BITS.
739 004630 016767 177672 177666 MOV PTIP,PTOP ;SAVE BIN CHAR IN PTO.
740 004636 016701 177664 MOV PTIP,%1 ;BIN CHAR TO R1.
741 004642 000207 RTS %7 ;EXIT.
742 :OCTAL TO ASCII CONVERT ROUTINES
743 004644 012500 ACNV6: MOV (5)+,%0 ;CONVERT TO 6 ASCII. GET OCTAL ADDRESS
744 004646 012567 000012 MOV (5)+,ACNVB ;GET ASCII ADDRESS
745 004652 004767 000052 JSR %7,ACNV ;CONVERT TO ASCII
746 004656 004567 000122 JSR %5,BMOVE ;MOVE 6 CHARS TO ASCII ADDRESS
747 004662 004720 AIST
748 004664 000000 ACNVB: OPEN
749 004666 000006 6
750 004670 000205 RTS %5 ;EXIT
751 004672 012500 ACNV4: MOV (5)+,%0 ;CONVERT TO 4 ASCII. GET OCTAL ADDRESS
752 004674 012567 000012 MOV (5)+,ACNVC ;GET ASCII ADDRESS
753 004700 004767 000024 JSR %7,ACNV ;CONVERT TO ASCII
754 004704 004567 000074 JSR %5,BMOVE ;MOVE 4 CHARS TO ASCII ADDRESS.
755 004710 004722 AIST+2
756 004712 000000 ACNVC: OPEN
757 004714 000004 4
758 004716 000205 RTS %5 ;EXIT
759 004720 000000 AIST: OPEN
760 004722 000000 OPEN
761 004724 000000 OPEN
762 004726 000000 ACNVX: OPEN
763 004730 012701 004726 ACNV: MOV #AIST+6,%1 ;ADDR TO STORE ASCII TO R1
764 004734 012702 000006 MOV #6,%2 ;6 TO R2
765 004740 011067 177762 MOV @%0,ACNVX ;OCTAL WORD TO ACNVX
766 004744 016703 177756 ACNVM: MOV ACNVX,%3
767 004750 042703 177770 BIC #177770,%3 ;ISOLATE LEAST SIGNIFICANT OCTAL #
768 004754 062703 000060 ADD #60,%3 ;ADD 60 TO CONVERT TO ASCII
769 004760 110341 MOVB %3,-(1) ;STORE ASCII BYTE
770 004762 006067 177740 ROR ACNVX ;MOVE NEXT OCTAL DIGIT TO LEAST
771 004766 006067 177734 ROR ACNVX ;SIGNIFICANT POSITION
772 004772 006067 177730 ROR ACNVX
773 004776 005302 DEC %2 ;DONE 6 TIMES?
774 005000 001361 BNE ACNVM ;NO. REPEAT.
775 005002 000207 RTS %7 ;YES. EXIT.

```

```

776
777
778 005004 104020
779 005006 012501
780 005010 012502
781 005012 012503
782 005014 112122
783 005016 005303
784 005020 001375
785 005022 104021
786 005024 000205
787
788 005026 105777 174064
789 005032 100001
790 005034 000207
791 005036 104010
792 005040 000772
793
794 005042 004767 177760
795 005046 004767 177012
796 005052 005000
797 005054 000207
798
799 005056 012700 017434
800 005062 013501
801 005064 012702 005164
802 005070 012767 000005 000060
803 005076 012267 000060
804 005102 004767 000010
805 005106 005367 000044
806 005112 001371
807 005114 000205
808 005116 005067 000036
809 005122 166701 000034
810 005126 103403
811 005130 005267 000024
812 005134 000772
813 005136 066701 000020
814 005142 062767 000060 000010
815 005150 116720 000004
816 005154 000207
817 005156 000000
818 005160 000000
819 005162 000000
820 005164 023420
821 005166 001750
822 005170 000144
823 005172 000012
824 005174 000001

```

```

;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
BMOVE: SAVREG ;SAVE REGS.
        MOV (5)+,%1 ;GET"FROM"ADDRESS
        MOV (5)+,%2 ;GET"TO"ADDRESS
        MOV (5)+,%3 ;GET COUNT
BMOVA: MOVB (1)+,(2)+ ;MOVE BYTE
        DEC %3 ;DECREMENT COUNT
        BNE BMOVA ;BRANCH IF NOT DONE.
        RSTREG ;RESTORE REGS.
        RTS %5 ;DONE EXIT

;SUBROUTINE TO CHECK FOR PUNCH READY.
CPRDY: TSTB %TPS ;TEST FOR READY BIT.
        BPL CPRDYA ;BRANCH IF READY NOT SET.
        RTS %7 ;OK. EXIT.
CPRDYA: EHALT ;NOT READY. HALT.
        BR CPRDY

;SUBROUTINE TO PUNCH ON LSP CHARACTER IN REG D.
LSPCH: JSR %7,CPRDY ;GO CHECK FOR PUNCH READY
        JSR %7,OUT ;GO TO OUTPUT ROUTINE.
        CLR %D
        RTS %7 ;DONE. EXIT.

;BINARY TO DECIMAL ASCII CONVER* SUBROUTINE.
BDCNV: MOV #DECVAL,%D ;SET UP ADDR TO STORE DECIMAL ASCII IN RD
        MOV @(%5)+,%1 ;BINARY VALUE TO R1.
        MOV #ADTEMP,%2 ;ADDR OF TEN POWER STRING TO R2.
        MOV #5,CNVCTR ;SET UP FOR 5 POWER CONVERSIONS.
BDCNVA: MOV (2)+,TENPWR ;MOVE POWER OF TEN VALUE TO TENPWR.
        JSR %7,SUBTEN ;PERFORM CONVERSION
        DEC CNVCTR ;DONE 5 CONVERSIONS?
        BNE BDCNVA ;BRANCH IF NOT YET 5.
        RTS %5 ;YES. EXIT.
SUBTEN: CLR DIGIT ;CLEAR DIGIT
SUBTNA: SUB TENPWR,%1 ;SUBTRACT TEN POWER FROM BINARY VALUE.
        BCS SUBTNB ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
        INC DIGIT
        BR SUBTNA
SUBTNB: ADD TENPWR,%1 ;RESTORE SUBTRACTED VALUE.
        ADD #60,DIGIT ;CONVERT (DIGIT) TO ASCII
        MOVB DIGIT,(0)+ ;MOVE ASCII CHAR TO DECVAL FIELD.
        RTS %7 ;EXIT.
CNVCTR: OPEN
DIGIT: OPEN
TENPWR: OPEN
ADTEMP: !0000.
        1000.
        100.
        10.
        1

```



```

825
826
827 005176 012767 000112 000050 :SUBROUTINE TO TYPE A LINE OF CHARACTERS
828 005204 012704 017441 TYPLN: MOV #74,TCTR ;72 TO CHAR COUNT +CR,LF
829 005210 104002 TYPLA: MOV #BLOCKA,%4 ;SET LINE ADDRESS IN R4.
830 005212 112400 TYPLB: STALL ;STALL IF ALLOWED.
831 005214 004767 177622 MOVB (4)+,%0 ;GET CHARACTER
832 005220 005367 000030 JSR %7,LSPCH ;GO OUTPUT CHARACTER.
833 005224 001371 DEC TCTR ;DONE?
834 005226 112767 000000 177052 TYPCLR: BNE TYPLB ;BRANCH IF NOT DONE.
835 005234 004767 176630 MOVB #0,TYPDAT
836 005240 112767 000000 177040 JSR %7,OUTTYP
837 005246 004767 176616 MOVB #0,TYPDAT
838 005252 000207 JSR %7,OUTTYP
839 005254 000000 RTS %7 ;DONE. EXIT
840
841 005256 011667 000016 :SUBROUTINE TO TYPE LINE OF 3 CHARACTERS
842 005262 017767 000012 000010 TYPL3: MOV @%6,TPL3A ;DEVELOP AND SET ADDRESS OF
843 005270 062716 000002 MOV @TPL3A,TPL3A ;DATA IN TPL3A.
844 005274 004567 000034 ADD #2,@%6 ;SET UP EXIT.
845 005300 000000 JSR %5,FBF3 ;FILL BUFFER WITH 3 CHARACTERS
846 005302 042767 040000 173642 TPL3A: OPEN
847 005310 004767 177662 BIC #BIT14,PRGID ;DISABLE STALLS.
848 005314 000002 JSR %7,TYPLN ;GO TYPE LINE OF CHARACTERS.
849 005316 112767 000015 012115 STBF: RTI ;EXIT
850 005324 112767 000012 012110 MOVB #15,BLOCKA ;SUB TO SET UP BUFFER AREA.
851 005332 000207 MOVB #12,BLOCKA+1
852
853 005334 012567 000004 :SUBROUTINE TO FILL CHARACTER BUFFER WITH 3 CHARACTERS.
854 005340 004567 177440 FBF3: MOV (5)+,FBF3A
855 005344 000000 JSR %5,BMOVE ;MOVE 3 CHARS TO BUFFER.
856 005346 017443 FBF3A: OPEN
857 005350 000003 BLOCK1
858 005352 004567 177426 FBF3B: 3
859 005356 017443 JSR %5,BMOVE ;FILL 72 CHARACTERS BUFFER
860 005360 017446 BLOCK1 ;WITH 3 CHARACTERS
861 005362 000105 BLOCK1+3
862 005364 004567 177414 69.
863 005370 017443 JSR %5,BMOVE
864 005372 017555 BLOCK1
865 005374 000110 BLOCK2
866 005376 000205 72.
RTS %5 ;EXIT

```

```

867
868
869 005400 004567 177400 ;SUBROUTINE TO FILL BUFFER WITH ALL CHARACTERS
870 005404 016466 FBALL: JSR %5,BMOVE ;FILL 72 CHAR BUFFER WITH
871 005406 017443 A ;ALL CHARACTERS.
872 005410 000077 BLOCK1
873 005412 004567 177366 63.
874 005416 016466 JSR %5,BMOVE
875 005420 017542 A
876 005422 000011 BLOCK1+63.
877 005424 004567 177354 9.
878 005430 017443 JSR %5,BMOVE
879 005432 017555 BLOCK1
880 005434 000110 BLOCK2
881 005436 000207 72.
892 ;SUB TO FILL BUFFER WITH 33 WORST CASE PATTERN. ;EXIT.
893 005440 004567 177340 FW336: JSR %5,BMOVE ;6 CHARACTER PATTERN TO BUFFER
894 005444 016452 A33WP6
895 005446 017443 BLOCK1
896 005450 000006 6
897 005452 004567 177326 JSR %5,BMOVE ;FILL BUFFER WITH PATTERN.
898 005456 017443 BLOCK1
899 005460 017451 BLOCK1+6
900 005462 000102 66.
901 005464 004567 177314 JSR %5,BMOVE
902 005470 017443 BLOCK1
903 005472 017555 BLOCK2
904 005474 000110 72.
905 005476 000207 RTS %7 ;EXIT
896
897 ;SUB TO FILL BUFFER WITH 35 WORST CASE PATTERN.
898 005500 004567 177300 FW356: JSR %5,BMOVE ;6 CHARACTER PATTERN TO BUFFER
899 005504 016460 A35WP6
900 005506 017443 BLOCK1
901 005510 000006 6
902 005512 004567 177266 JSR %5,BMOVE ;FILL BUFFER WITH PATTERN.
903 005516 017443 BLOCK1
904 005520 017451 BLOCK1+6
905 005522 000102 66.
906 005524 004567 177254 JSR %5,BMOVE
907 005530 017443 BLOCK1
908 005532 017555 BLOCK2
909 005534 000110 72.
910 005536 000207 RTS %7 ;EXIT.

```

```

911
912
913 005540 012767 005552 173370 :PRGO - INPUT-OUTPUT LOGIC TESTS
914 005546 000167 174436 PRGO: MOV #ATO,KSTART ;ADDRESS OF 1ST ROUTINE TO KSTART.
          JMP SRSET ;GO GET STARTED.
915 :TEST ABILITY TO REFERENCE THE KEYBOARD/READER STATUS WORD (TKS)
916 005552 000000 ATO: 0 ;TEST #.
917 005554 005604 AT1 ;NEXT TEST.
918 005556 001750 1000. ;I COUNT.
919 005560 005570 ATOA ;SCOPE ENTRY.
920 005562 012767 005600 172214 MOV #ATOE,MACHER ;SET UP MACHINE ERROR TRAP.
921 005570 005777 173316 ATOA: TST @TKS ;REFERENCE CODER STATUS WORD.
922 005574 104012 ATOB: CHAIN ;CHAIN
923 005576 000774 BR ATOA ;REPEAT TEST.
924 005600 104003 ATOE: ERROR ;ERROR. TRAPPED WHEN REFERENCING READER.
925 005602 000774 BR ATOB ;STATUS WORD (TKS).
926 :TEST ABILITY TO REFERENCE THE KEYBOARD/READER BUFFER (TKB).
927 005604 000001 AT1: 1 ;TEST #.
928 005606 005636 AT2 ;NEXT TEST.
929 005610 001750 1000. ;I COUNT.
930 005612 005622 AT1A ;SCOPE ENTRY.
931 005614 012767 005632 172162 MOV #AT1E,MACHER ;SET UP MACHINE ERROR TRAP
932 005622 005777 173266 AT1A: TST @TKB ;REFERENCE READER BUFFER.
933 005626 104012 AT1B: CHAIN ;CHAIN
934 005630 000774 BR AT1A ;REPEAT TEST.
935 005632 104003 AT1E: ERROR ;ERROR. TRAPPED WHEN REFERENCING
936 005634 000774 BR AT1B ;READER BUFFER. (TKB).
937 :TEST ABILITY TO REFERENCE PUNCH/PRINTER STATUS WORD (TPS).
938 005636 000002 AT2: 2 ;TEST #.
939 005640 005670 AT3 ;NEXT TEST
940 005642 001750 1000. ;I COUNT.
941 005644 005654 AT2A ;SCOPE ENTRY.
942 005646 012767 005664 172130 MOV #AT2E,MACHER ;SETUP MACHINE ERROR TRAP.
943 005654 005777 173236 AT2A: TST @TPS ;REFERENCE PUNCH/PRINTER STATUS WORD.
944 005660 104012 AT2B: CHAIN ;CHAIN
945 005662 000774 BR AT2A ;REPEAT TEST.
946 005664 104003 AT2E: ERROR ;ERROR. TRAPPED WHEN REFERENCING
947 005666 000774 BR AT2B ;PUNCH/PRINTER STATUS WORD (TPS).
948 :TEST ABILITY TO REFERENCE PUNCH/PRINTER BUFFER (TPB).
949 005670 000003 AT3: 3
950 005672 005722 AT4
951 005674 001750 1000.
952 005676 005706 AT3A
953 005700 012767 005716 172076 MOV #AT3E,MACHER ;SETUP MACHINE ERROR TRAP.
954 005706 005777 173206 AT3A: TST @TPB ;REFERENCE PUNCH/PPINTER BUFFER.
955 005712 104012 AT3B: CHAIN ;CHAIN
956 005714 000774 BR AT3A ;REPEAT TEST.
957 005716 104003 AT3E: ERROR ;ERROR. TRAPPED WHEN REFERENCING
958 005720 000774 BR AT3B ;PUNCH/PRINTER BUFFER. (TPS).
  
```

```

959
960
961 005722 000004
962 005724 006006
963 005726 001750
964 005730 005740
965 005732 012767 000340 172036
966 005740 052777 000100 173144
967 005746 032777 000100 173136
968 005754 001002
969 005756 104003
970 005760 000410
971 005762 042777 000100 173122
972 005770 032777 000100 173114
973 005776 001401
974 006000 104003
975 006002 104012
976 006004 000755
977
978 006006 000005
979 006010 006676
980 006012 000144
981 006014 006024
982 006016 012767 000340 171752
983 006024 052777 000100 173060
984 006032 104011
985 006034 032777 000100 173050
986 006042 001401
987 006044 104003
988 006046 104012
989 006050 000765
990
991 006052 000011
992 006054 006076
993 006056 000144
994 006060 006062
995 006062 104400
996 006064 000226
997 006066 004767 175452
998 006072 104012
999 006074 000772

;TEST ABILITY TO SET AND CLEAR READER/KYBD ID BIT
AT4: 4
      AT5
      1000.
      AT4A
      MOV #PRTY7,PSW
      BIS #BIT6,@TKS
      BIT #BIT6,@TKS
      BNE AT4B
AT4E1: ERROR
      BR AT4C
AT4B: BIC #BIT6,@TKS
      BIT #BIT6,@TKS
      BEQ AT4C
AT4E2: ERROR
AT4C: CHAIN
      BR AT4A
;TEST ABILITY TO CLEAR ID BIT WITH RESET INSTRUCTION.
AT5: 5
      AT24
      100.
      AT5A
      MOV #PRTY7,PSW
      BIS #BIT6,@TKS
      SRESET
      BIT #BIT6,@TKS
      BEQ AT5B
AT5E: ERROR
AT5B: CHAIN
      BR AT5A
;TEST THAT READER DONE BIT SETS BY 200 MSECS AFTER READER ENABLE
AT11: !1
      AT12
      100.
      AT11A
      DELAY
      150.
      JSR %7,AREAD
AT11B: CHAIN
      BR AT11A

;TEST #
;NEXT TEST
;I COUNT
;SCOPE ENTRY
;SET PRIORITY 7.
;SET ID BIT IN TKS.
;CHECK ID BIT IN TKS
;BRANCH IF ID BIT IS SET.
;ERROR 1 ID BIT NOT SET.

;CLEAR ID BIT IN TKS
;CHECK ID BIT IN TKS.
;BRANCH IF ID BIT IS CLEARED.
;ERROR. ID BIT FAILED TO CLEAR.
;CHAIN
;REPEAT TEST.

;TEST #
;NEXT TEST
;I COUNT
;SCOPE ENTRY.
;SET PRIORITY 7.
;SET ID BIT IN TKS
;RESET
;TEST ID BIT.
;BRANCH IF ID BIT IS CLEAR.
;ERROR. RESET FAILED TO CLEAR ID BIT.
;CHAIN
;REPEAT TEST.

;WAIT 150 MSECS
;ENABLE READER
;CHAIN
;REPEAT TEST

```

```

1000
1001 006076 000012
1002 006100 006150
1003 006102 000012
1004 006104 006106
1005 006106 104400
1006 006110 000226
1007 006112 004767 175426
1008 006116 012767 001750 173140
1009 006124 105777 172762
1010 006130 100402
1011 006132 104003
1012 006134 000403
1013 006136 005367 173122
1014 006142 001370
1015 006144 104012
1016 006146 000761
1017
1018 006150 000013
1019 006152 006206
1020 006154 000144
1021 006156 006160
1022 006160 104400
1023 006162 000226
1024 006164 004767 175354
1025 006170 104011
1026 006172 105777 172714
1027 006176 100001
1028 006200 104003
1029 006202 104012
1030 006204 000767
1031
1032 006206 000014
1033 006210 006246
1034 006212 000144
1035 006214 006216
1036 006216 104400
1037 006220 000226
1038 006222 004767 175316
1039 006226 105777 172662
1040 006232 105777 172654
1041 006236 100001
1042 006240 104003
1043 006242 104012
1044 006244 000766
1045
1046 006246 000017
1047 006250 006342
1048 006252 000144
1049 006254 006256
1050 006256 104400
1051 006260 000226
1052 006262 004767 175256
1053 006266 117767 172622 172757
1054 006274 012767 000144 172762
1055 006302 117767 172606 172742

;TEST THAT DONE BIT READS RELIABLY.
AT12: 12 ;TEST #
      AT13 ;NEXT TEST
      10. ;I COUNT
      AT12A ;SCOPE ENTRY.
      DELAY ;WAIT 150 MSECS.
      150.
AT12B: JSR %7,AREAD ;ENABLE READER. COME BACK WHEN DONE SET.
      MOV #1000.,CTRA ;1000 TO CTRA
AT12C: TSTB @TKS ;TEST FOR DONE
      BMI AT12D ;BRANCH IF DONE FOUND SET.
AT12E: ERROR ;ERROR. DONE BIT NOT FOUND SET.
      BR AT12F
AT12D: DEC CTRA ;CHECKED 1000 TIMES?
      BNE AT12C ;BRANCH IF NOT 1000 YET.
AT12F: CHAIN ;CHAIN
      BR AT12B ;REPEAT TEST.

;TEST THAT RESET CLEARS DONE BIT
AT13: 13 ;TEST #
      AT14 ;NEXT TEST
      100. ;I COUNT
      AT13A ;SCOPE ENTRY
      DELAY ;WAIT 150 MSECS.
      150.
AT13B: JSR %7,AREAD ;ENABLE READER. COME BACK WHEN DONE SET.
      SRESET ;ISSUE RESET.
      TSTB @TKS ;TEST FOR DONE BIT
      BPL AT13C ;BRANCH IF DONE BIT RESET.
AT13E: ERROR ;ERROR. RESET FAILED TO CLEAR DONE.
AT13C: CHAIN ;CHAIN
      BR AT13B ;REPEAT TEST.

;TEST THAT REFERENCING READER DATA BUFFER CLEARS DONE
AT14: 14 ;TEST #
      AT17 ;NEXT TEST
      100. ;I COUNT
      AT14A ;SCOPE ENTRY
      DELAY ;WAIT 150 MSECS.
      150.
AT14B: JSR %7,AREAD ;ENABLE READER. RETURN WHEN DONE SET.
      TSTB @TKB ;REFERENCE READ BUFFER.
      TSTB @TKS ;TEST FOR DONE BIT
      BPL AT14C ;BRANCH IF DONE NOT SET.
AT14E: ERROR ;ERROR. REFERENCE TO BUFFER DID NOT RESET DONE.
AT14C: CHAIN ;CHAIN
      BR AT14B ;REPEAT TEST

;TEST THAT READ BUFFER CAN BE READ RELIABLY.
AT17: 17 ;TEST #
      AT20 ;NEXT TEST
      100. ;I COUNT
      AT17A ;SCOPE ENTRY
      DELAY ;WAIT 150 MSECS
      150.
AT17B: JSR %7,AREAD ;ENABLE READER. RETURN WHEN DONE SET.
      MOV @TKB,CRBUF+1 ;BUFFER CONTENTS TO CRBUF+1
      MOV #100.,CTRA ;100 TO CTRA
AT17C: MOV @TKB,CRBUF ;BUFFER CONTENTS TO CRBUF

```

006310	126767	172736	172735
006316	007404		
006320	016700	172726	
006324	000000		
006326	000403		
006330	005367	172730	
006334	001362		
006336	104012		
006340	000750		

	CMPB	CRBUF,CRBUF+1
	BEQ	AT17D
	MOV	CRBUF,%0
AT17E:	HALT	
	BR	AT17F
AT17D:	DEC	CTRA
	BNE	AT17C
AT17F:	CHAIN	
	BR	AT17B

```

:COMPARE CONTENTS OF CRBUF AND CRBUF+1
:BRANCH IF SAME.
:NOT SAME. ERROR. HALT WITH 1ST READ CHAR
:IN DATA BYTES LEFT. SUBSEQUENT READ IN DATA BYTES RIGHT
:HERE IF SAME. CHECKED 100 TIMES?
:BRANCH IF NOT.
:CHAIN
:REPEAT TEST.

```

```

1065
1066
1067
1068 006342 000020
1069 006344 006420
1070 006346 001750
1071 006350 006366
1072 006352 104006
1073 006354 006412
1074 006356 104400
1075 006360 000226
1076 006362 004767 175156
1077 006366 005077 172520
1078 006372 005067 171400
1079 006376 052777 000100 172506
1080 006404 000240
1081 006406 104003
1082 006410 000401
1083 006412 022626
1084 006414 104012
1085 006416 000763
1086
1087
1088 006420 000021
1089 006422 006504
1090 006424 001750
1091 006426 006444
1092 006430 104006
1093 006432 006476
1094 006434 104400
1095 006436 000226
1096 006440 004767 175100
1097 006444 005077 172442
1098 006450 016767 172450 171320
1099 006456 052777 000100 172426
1100 006464 000240
1101 006466 005077 172420
1102 006472 104012
1103 006474 000763
1104 006476 022626
1105
1106 006500 104003
1107 006502 000771
1108
1109
1110
1111
1112
1113 006504 000022
1114 006506 006576
1115 006510 001750
1116 006512 006564
1117 006514 104006
1118 006516 006564
1119 006520 104400
1120 006522 000226

```

```

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

```

```

AT20: 20 ;TEST#
      AT21 ;NEXT TEST
      1000. ;I COUNT
      AT20B ;SCOPE ENTRY
      STRDRV ;SET UP READER VECTOR TO AT20C
      AT20C
AT20A: DELAY ;WAIT 150 MSECS.
      150.
      JSR %7,AREAD ;ENABLE READER. RETURN WHEN DONE SET.
AT20B: CLR @TKS ;DISABLE READER INTERRUPTS
      CLR PSW ;ENABLE READER. RETURN WHEN DONE SET.
      BIS #BIT6,@TKS ;ENABLE READER INTERRUPT,
      NOP
AT20E: ERROR ;ERROR. READER FAILED TO INTERRUPT.
      BR AT20D
AT20C: POPSP2 ;HERE IF INTERRUPT OCCURS. POP STACK TWICE.
AT20D: CHAIN ;CHAIN
      BR AT20B ;REPEAT TEST.
:TEST THAT DONE DOES NOT CAUSE INTERRUPT WITH PROCESSOR AT SAME
PRIORITY LEVEL AS THE READERS INTERRUPT REQUEST LEVEL.
AT21: 21 ;TEST#
      AT22 ;NEXT TEST.
      1000. ;I COUNT.
      AT21B ;SCOPE ENTRY.
      STRDRV ;SET READER VECTOR TO AT21E.
      AT21E
AT21A: DELAY ;WAIT 150 MSECS.
      150.
      JSR %7,AREAD ;ENABLE READER. RETURN WHEN DONE SET.
AT21B: CLR @TKS ;DISABLE READER INTERRUPTS.
      MOV TKLVL,PSW ;SET PROCESSOR TO SAME PRIORITY AS READER'S.
      BIS #BIT6,@TKS ;ENABLE READER INTERRUPTS.
      NOP ;NO OP.
AT21C: CLR @TKS ;OK IF NO INTERRUPT OCCURS.
      CHAIN ;CHAIN
      BR AT21B ;REPEAT TEST.
AT21E: POPSP2 ;ERROR. READER ERRONEOUSLY INTERRUPTED
      ERROR ;WITH PROCESSOR AT SAME PRIORITY
      BR AT21C ;LEVEL AS THE READER, OR THE READER
      ;IS AT HIGHER PRIORITY THAN SPECIFIED
      ;AT TKLVL.

```

```

:TEST THAT DONE CAUSES INTERRUPT WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
THAN THE READER'S INTERRUPT PRIORITY LEVEL.

```

```

AT22: 22 ;TEST#
      AT23 ;NEXT TEST
      1000. ;I COUNT
      AT22B ;SCOPE ENTRY
      STRDRV ;SET READER INTERRUPT SERVICE TO
      AT22B ;AT22B.
      DELAY ;WAIT 150 MSECS.
      150.

```

```

1121 006524 004767 175014          JSR      %7,AREAD      ;ENABLE READER. RETURN WHEN DONE SET.
1122 006530 005077 172356          CLR      %TKS         ;DISABLE READER INTERRUPTS
1123 006534 016767 172364 171234      MOV      TKLVL,PSW    ;SET PROCESSOR PRIORITY ONE LEVEL LOWER
1124 006542 162767 000040 171226      SUB      #40,PSW     ;THAN READER.(SPECIFIED AT TKLVL).
1125 006550 052777 000100 172334      BIS      #BIT6,%TKS  ;ENABLE READER INTERRUPTS.
1126 006556 000240          NOP
1127 006560 104003          AT22E:  ERROR        ;FAILED TO INTERRUPT WITH PC AT PRIORITY ONE LEVEL LOWER
1128 006562 000401          BR      AT22C        ;THAN READER. THEREFORE, READER PRIORITY MUST BE LOWER
1129 006564 022626          AT22B:  POPSP2       ;HERE IF INTERRUPT OCCURS. OK. POP STACK TWICE
1130 006566 005077 172320          AT22C:  CLR      %TKS  ;DISABLE READER INTERRUPTS
1131 006572 104012          CHAIN
1132 006574 000755          BR      AT22A        ;REPEAT TEST
1133          ;TEST THAT DONE DOES NOT REINTERRUPT AFTER RTI WHEN DONE IS NOT CLEARED.
1134 006576 000023          AT23:   23          ;TEST#
1135 006600 010046          AT43          ;NEXT TEST
1136 006602 001750          1000.        ;I COUNT
1137 006604 006616          AT23A       ;SCOPE ENTRY.
1138 006606 104400          DELAY        ;WAIT 150 MSECS
1139 006610 000226          150.
1140 006612 004767 174726          JSR      %7,AREAD    ;ENABLE READER. RETURN WHEN DONE SET.
1141 006616 104006          AT23A:  STRDRV      ;SET READER INTERRUPT SERVICE
1142 006620 006650          AT23C       ;TO AT23C.
1143 006622 005077 172264          CLR      %TKS        ;DISABLE READER INTERRUPTS.
1144 006626 052777 000100 172256      BIS      #BIT6,%TKS  ;ENABLE READER INTERRUPTS.
1145 006634 000240          NOP
1146 006636 104003          AT23E1:  ERROR      ;ERROR1 FAILED TO INTERRUPT
1147 006640 005077 172246          AT23B:  CLR      %TKS  ;DISABLE READER INTERRUPTS.
1148 006644 104012          CHAIN
1149 006646 000763          BR      AT23A        ;REPEAT TEST
1150 006650 012777 006670 172244          AT23C:  MOV      #AT23E2,%TKVTR ;CHANGE INTERRUPT VECTOR TO AT23E2
1151 006656 012716 006664          MOV      #AT23D,%6
1152 006662 000002          RTI          ;RETURN FROM INTERRUPT
1153 006664 000240          AT23D:  NOP
1154 006666 000764          BR      AT23B        ;OK IF NO REINTERRUPT OCCURS.
1155 006670 022626          AT23E2:  POPSP2     ;ERROR2. DONE REINTERRUPTED AFTER
1156 006672 104003          ERROR      ;RTI WITH DONE BIT LEFT ON.
1157 006674 000761          BR      AT23B
1158          ;TEST ABILITY TO SET AND CLEAR PUNCH ID BIT
1159 006676 000024          AT24:   24          ;TEST#
1160 006700 006762          AT25          ;NEXT TEST.
1161 006702 001750          1000.        ;I COUNT
1162 006704 006714          AT24A       ;SCOPE ENTRY.
1163 006706 012767 000340 171062          MOV      #PRTY7,PSW ;SET PRIORITY 7.
1164 006714 052777 000100 172174          AT24A:  BIS      #BIT6,%TPS  ;SET PUNCH ID BIT.
1165 006722 032777 000100 172166          BIT      #BIT6,%TPS  ;CHECK PUNCH ID BIT.
1166 006730 001002          BNE      AT24B       ;BRANCH IF PUNCH ID BIT IS SET.
1167 006732 104003          AT24E1:  ERROR      ;ERROR1. PUNCH ID BIT DID NOT SET.
1168 006734 000410          BR      AT24C
1169 006736 042777 000100 172152          AT24B:  BIC      #BIT6,%TPS  ;CLEAR PUNCH ID BIT.
1170 006744 032777 000100 172144          BIT      #BIT6,%TPS  ;CHECK PUNCH ID BIT.
1171 006752 001401          BEQ      AT24C       ;BRANCH IF PUNCH ID BIT IS CLEAR
1172 006754 104003          AT24E2:  ERROR      ;ERROR2. PUNCH ID BIT FAILED TO CLEAR.
1173 006756 104012          AT24C:  CHAIN
1174 006760 000776          BR      AT24C        ;REPEAT TEST
1175          ;TEST ABILITY TO CLEAR PUNCH ID BIT WITH RESET INSTRUCTION
1176 006762 000025          AT25:   25          ;TEST#

```


1177	006764	007026				AT26			:NEXT TEST.
1178	006766	000144				100			:I COUNT.
1179	006770	007000				AT25A			:SCOPE ENTRY.
1180	006772	012767	000340	170776		MOV	#PRTY7,PSW		:SET PRIORITY 7.
1181	007000	052777	000100	172110	AT25A:	BIS	#BIT6,@TPS		:SET PUNCH ID BIT.
1182	007006	104011				SRESET			:RESET
1183	007010	032777	000100	172100		BIT	#BIT6,@TPS		:CHECK PUNCH ID BIT.
1184	007016	001401				BEQ	AT25B		:BRANCH IF PUNCH ID BIT IS CLEAR.
1185	007020	104003			AT25E:	ERROR			:ERROR. RESET FAILED TO CLEAR PUNCH ID BIT.
1186	007022	104012			AT25B:	CHAIN			:CHAIN
1187	007024	000765				BR	AT25A		:REPEAT TEST.
1188						:TEST ABILITY TO SET AND CLEAR THE PUNCH MAINTENANCE BIT			
1189	007026	000026			AT26:	26			:TEST*
1190	007030	007104				AT27			:NEXT TEST
1191	007032	001750				1000.			:I COUNT
1192	007034	007036				AT26A			:SCOPE ENTRY
1193	007036	052777	000004	172052	AT26A:	BIS	#BIT2,@TPS		:SET MAINTANCE BIT.
1194	007044	032777	000004	172044		BIT	#BIT2,@TPS		:CHECK MAINTENANCE BIT
1195	007052	001002				BNE	AT26B		:BRANCH IF MAINTENANCE BIT SET.
1196	007054	104003			AT26E1:	ERROR			:ERROR1. MAINTENANCE BIT FAILED TO SET.
1197	007056	000410				BR	AT26C		
1198	007060	042777	000004	172030	AT26B:	BIC	#BIT2,@TPS		:CLEAR MAINTENANCE BIT.
1199	007066	032777	000004	172022		BIT	#BIT2,@TPS		:CHECK MAINTENANCE BIT
1200	007074	001401				BEQ	AT26C		:BRANCH IF MAINTENANCE BIT IS CLEAR.
1201	007076	104003			AT26E2:	ERROR			:ERROR2 MAINTENANCE BIT FAILED TO CLEAR.
1202	007100	104012			AT26C:	CHAIN			:CHAIN
1203	007102	000755				BR	AT26A		:REPEAT TEST

```

1204
1205      :TEST THAT RESET INSTRUCTION CLEARS THE MAINTENANCE BIT.
1206 007104 000027 AT27: 27      :TEST#
1207 007106 007142      AT30      :NEXT TEST
1208 007110 000144      100.      :I COUNT
1209 007112 007114      AT27A     :SCOPE ENTRY
1210 007114 052777 000004 171774 AT27A: BIS      #BIT2,@TPS :SET MAINTENANCE BIT.
1211 007122 104011      SRESET   :ISSUE RESET
1212 007124 032777 000004 171764      BIT      #BIT2,@TPS :CHECK MAINTENANCE BIT
1213 007132 001401      BEQ      AT27B     :BRANCH IF MAINTENANCE BIT CLEAR.
1214 007134 104003      AT27E: ERROR   :ERROR. RESET FAILED TO CLEAR
1215 007136 104012      AT27B: CHAIN   :THE MAINTENANCE BIT. CHAIN.
1216 007140 000765      BR      AT27A     :REPEAT TEST.
1217      :TEST THAT RESET SETS THE PUNCH READY BIT, AND THAT READY CAN BE READ RELIABLY.
1218 007142 000030 AT30: 30      :TEST#
1219 007144 007166      AT31      :NEXT TEST
1220 007146 001750      1000.     :I COUNT
1221 007150 007152      AT30A     :SCOPE ENTRY
1222 007152 105777 171740 AT30A: TSTB    @TPS    :CHECK PUNCH READY.
1223 007156 100401      BMI      AT30B     :BRANCH IF PUNCH READY IS SET.
1224 007160 104003      AT30E: ERROR   :ERROR. RESET FAILED TO SET READY, OR FAILED TO READ IT
1225 007162 104012      AT30B: CHAIN   :CHAIN
1226 007164 000772      BR      AT30A     :REPEAT TEST.
1227      :TEST THAT PUNCH READY RESETS BY LOADING PUNCH BUFFER.
1228 007166 000031 AT31: 31      :TEST#
1229 007170 007224      AT32      :NEXT TEST
1230 007172 000024      20.      :I COUNT
1231 007174 007176      AT31A     :SCOPE ENTRY
1232 007176 104400      AT31A: DELAY   :WAIT 150 MSECS
1233 007200 000226      150.
1234 007202 104011      SRESET   :RESET
1235 007204 005077 171710      CLR      @TPB     :LOAD PUNCH BUFFER
1236 007210 105777 171702      TSTB    @TPS     :CHECK PUNCH READY BIT.
1237 007214 100001      BPL      AT31B     :BRANCH IF PUNCH READY IS CLEAR.
1238 007216 104003      AT31E: ERROR   :ERROR. BUFFER LOAD FAILED TO CLEAR READY.
1239 007220 104012      AT31B: CHAIN   :CHAIN
1240 007222 000765      BR      AT31A     :REPEAT TEST.
1241      :TEST THAT BYTE LOAD OF PUNCH BUFFER +1 DOES NOT RESET READY.
1242 007224 000032 AT32: 32      :TEST#
1243 007226 007266      AT33      :NEXT TEST
1244 007230 000024      20.      :I COUNT
1245 007232 007234      AT32A     :SCOPE ENTRY
1246 007234 104400      AT32A: DELAY   :WAIT 150 MSECS
1247 007236 000226      150.
1248 007240 104011      SRESET   :RESET
1249 007242 016700 171652      MOV      TPB,%0
1250 007246 005200      INC      %0
1251 007250 105010      CLRB    @%0      :BYTE LOAD PUNCH BUFFER+1
1252 007252 105777 171640      TSTB    @TPS     :CHECK PUNCH READY BIT
1253 007256 100401      BMI      AT32B     :BRANCH IF PUNCH READY STILL SET.
1254 007260 104003      AT32E: ERROR   :ERROR. BYTE LOAD OF PUNCH BUFFER+1
1255 007262 104012      AT32B: CHAIN   :CLEARED READY. CHAIN
1256 007264 000765      BR      AT32A     :REPEAT TEST.

```

G05

KL11 TESTS FOR VT05 DISPLAY TERMINAL
DZVT8.P11

MACY11 27(732) 20-SEP-76 10:11 PAGE 30

SEQ 0058

```

1257
1258
1259 007266 000033
1260 007270 007326
1261 007272 000024
1262 007274 007276
1263 007276 104400
1264 007300 000226
1265 007302 005077 171612
1266 007306 104400
1267 007310 000310
1268 007312 105777 171600
1269 007316 100401
1270 007320 104003
1271 007322 104012
1272 007324 000764
1273
1274
1275 007326 000034
1276 007330 007374
1277 007332 001750
1278 007334 007342
1279 007336 104007
1280 007340 007370
1281 007342 005077 171550
1282 007346 005067 170424
1283 007352 052777 000100 171536
1284 007360 000240
1285 007362 104003
1286 007364 104012
1287 007366 000765
1288 007370 022626
1289 007372 000774
1290
1291
1292 007374 000035
1293 007376 007450
1294 007400 001750
1295 007402 007410
1296 007404 104007
1297 007406 007442
1298 007410 016767 171514 170360
1299 007416 005077 171474
1300 007422 052777 000100 171466
1301 007430 000240
1302 007432 005077 171460
1303 007436 104012
1304 007440 000763
1305 007442 02626
1306 007444 104003
1307 007446 000771

;TEST THAT PUNCH BECOMES READY BY 200 MSECS AFTER BUFFER LOAD.
AT33: 33 ;TEST #
      AT34 ;NEXT TEST
      20. ;I COUNT
      AT33A ;SCOPE ENTRY.
      DELAY ;WAIT 150 MSECS.
      150.
      CLR @TPB ;LOAD PUNCH BUFFER.
      DELAY ;WAIT 200 MSECS.
      200.
      TSTB @TPS ;CHECK PUNCH READY BIT.
      BMI AT33B ;BRANCH IF PUNCH READY IS SET.
      AT33E: ERROR ;ERROR. READY NOT SET 200 MSECS AFTER BUFFER LOAD.
      AT33B: CHAIN ;CHAIN
      BR AT33A ;REPEAT TEST.

;TEST THAT PUNCH READY BIT CAN CAUSE AN INTERRUPT. IF THE INTERRUPT
;IS SERVICED, IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
AT34: 34 ;TEST #
      AT35 ;NEXT TEST
      1000. ;I COUNT
      AT34A ;SCOPE ENTRY
      STPCHV ;SET PUNCH INTERRUPT SERVICE
      AT34C ;TO AT34C
      CLR @TPS ;DISABLE PUNCH INTERRUPTS
      CLR PSW ;SET PRIORITY 0.
      BIS #BIT6,@TPS ;ENABLE PUNCH INTERRUPTS.
      NOP
      AT34E: ERROR ;PUNCH READY FAILED TO CAUSE
      AT34B: CHAIN ;INTERRUPT. CHAIN
      BR AT34A ;REPEAT TEST.
      AT34C: POPSP2 ;HERE IF INTERRUPT OCCURS. POP THE
      BR AT34B ;STOCK TWICE.

;TEST THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WITH PROCESSOR
;AT SAME PRIORITY LEVEL AS THE PUNCH INTERRUPT REQUEST LEVEL.
AT35: 35 ;TEST #
      AT36 ;NEXT TEST
      1000. ;I COUNT
      AT35A ;SCOPE ENTRY
      STPCHV ;SET PUNCH INTERRUPT SERVICE
      AT35E ;TO AT35E.
      MOV TPLVL,PSW ;SET PROCESSOR TO SAME PRIORITY AS PUNCH.
      CLR @TPS ;DISABLE PUNCH INTERRUPTS.
      BIS #BIT6,@TPS ;ENABLE PUNCH INTERRUPTS.
      NOP
      AT35B: CLR @TPS ;OK IF NO INTERRUPT OCCURS.
      CHAIN ;CHAIN
      BR AT35A ;REPEAT TEST.
      AT35E: POPSP2 ;ERROR. PUNCH INTERRUPTED WITH PROCESSOR
      ERROR ;SET TO SAVE PRIORITY AS THE PUNCH.
      BR AT35B

```

```

1308
1309
1310
1311 007450 000036
1312 007452 007532
1313 007454 001750
1314 007456 007464
1315 007460 104007
1316 007462 007523
1317 007464 005077 171426
1318 007470 016767 171434 170300
1319 007476 162767 000040 170272
1320 007504 052777 000100 171404
1321 007512 000240
1322 007514 104003
1323 007516 000401
1324 007520 022626
1325 007522 005077 171370
1326 007526 104012
1327 007530 000755
1328
1329
1330 007532 000037
1331 007534 007626
1332 007536 001750
1333 007540 007542
1334 007542 104007
1335 007544 007600
1336 007546 005077 171344
1337 007552 005067 170220
1338 007556 052777 000100 171332
1339 007564 000240
1340 007566 104003
1341 007570 005077 171322
1342 007574 104012
1343 007576 000761
1344 007600 012777 007620 171320
1345 007606 012716 007614
1346 007612 000002
1347 007614 000240
1348 007616 000764
1349 007620 022626
1350 007622 104003
1351 007624 000761

;TEST THAT THE PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
;THAN THE PUNCH PRIORITY.
AT36: 36
      AT37
      1000.
      AT36A
      STPCHV
      AT36B
AT36A: CLR @TPS
      MOV TPLVL,PSW
      SUB #40,PSW
      BIS #BIT6,@TPS
      NOP
      ERROR
      BR AT36C
AT36B: POPSP2
AT36C: CLR @TPS
      CHAIN
      BR AT36A

;TEST THAT PUNCH READY DOES NOT REINTERRUPT AFTER RTI WHEN READY
;BIT HAS NOT BEEN RESET.
AT37: 37
      AT40
      1000.
      AT37A
AT37A: STPCHV
      AT37C
      CLR @TPS
      CLR PSW
      BIS #BIT6,@TPS
      NOP
AT37E1: ERROR
AT37B: CLR @TPS
      CHAIN
      BR AT37A
AT37C: MOV #PT37E2,@TPVTR
      MOV #AT37D,@%6
      RTI
AT37D: NOP
      BR AT37B
AT37E2: POPSP2
      ERROR
      BR AT37B

;TEST #
;NEXT TEST
;I COUNT
;SCOPE ENTRY
;SET PUNCH INTERRUPT SERVICE
;TO AT36B.
;DISABLE PUNCH INTERRUPTS
;SET PROCESSOR PRIORITY ONE LEVEL
;LOWER THAN PUNCH PRIORITY
;ENABLE PUNCH INTERRUPTS
;ERROR. PUNCH FAILED TO INTERRUPT.
;HERE IF INTERRUPT OCCURS. POP
;THE STACK TWICE. DISABLE PUNCH INTERRUPT
;CHAIN
;REPEAT TEST.
;TEST #
;NEXT TEST
;I COUNT
;SCOPE ENTRY
;SET PUNCH INTERRUPT SERVICE TO
;AT37C
;DISABLE PUNCH INTERRUPTS
;SET PROCESSOR PRIORITY TO 0
;ENABLE PUNCH INTERRUPTS
;ERROR 1. PUNCH FAILED TO INTERRUPT.
;DISABLE PUNCH INTERRUPT.
;CHAIN
;REPEAT TEST.
;HERE IF INTERRUPT OCCURS. CHANGE
;PUNCH VECTOR TO AT37E2 AND EXIT
;INTERRUPT
;OK IF NO REINTERRUPT OCCURS
;ERROR 2. PUNCH REINTERRUPTED AFTER
;RTI WITH READY BIT LEFT ON

```

```

1352
1353
1354
1355 007626 000040
1356 007630 007712
1357 007632 001750
1358 007634 007642
1359 007636 104007
1360 007640 007700
1361 007642 012767 000340 170126 AT40A: MOV #PRTY7,PSW
1362 007650 005077 171242 CLR @TPS
1363 007654 052777 000100 171234 BIS #BIT6,@TPS
1364 007662 005067 170110 CLR PSW
1365 007666 012767 000340 170102 MOV #PRTY7,PSW
1366 007674 104003 ERROR
1367 007676 000401 BR AT40C
1368 007700 022626 AT40B: POPSP2
1369 007702 005077 171210 AT40C: CLR @TPS
1370 007706 104012 CHAIN
1371 007710 000754 BR AT40A
1372
1373
1374
1375
1376
1377
1378
1379
1380 007712 000041
1381 007714 010000
1382 007716 000062
1383 007720 007732
1384 007722 104400
1385 007724 000226
1386 007726 104007
1387 007730 007770
1388 007732 012767 000001 000016 AT41A: MOV #WAIT,AT41B
1389 007740 005077 171154 CLR @TPB
1390 007744 052777 000100 171144 BIS #BIT6,@TPS
1391 007752 005067 170020 CLR PSW
1392 007756 000000 AT41B: OPEN
1393
1394
1395
1396 007760 005077 171132 CLR @TPS
1397 007764 104012 CHAIN
1398 007766 000761 BR AT41A
1399 007770 012767 104003 177760 AT41C: MOV #ERROR,AT41B
1400 007776 000002 RTI

```

;TEST THAT THE PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING
;PROCESSOR PRIORITY TO 0.

```

;TEST #
;NEXT TEST
;I COUNT
;SCOPE ENTRY
;SET PUNCH INTERRUPT
;SERVICE TO AT40B
;SET PROCESSOR PRIORITY TO 7.
;DISABLE PUNCH INTERRUPTS
;ENABLE PUNCH INTERRUPTS
;LOWER PROCESSOR PRIORITY TO 0.
;RAISE PRIORITY TO 7.
;ERROR. PUNCH FAILED TO INTERRUPT
;IMMEDIATELY AFTER CP PRIORITY WAS SET TO 0.
;HERE IF INTERRUPT OCCURS
;DISABLE PUNCH INTERRUPTS
;CHAIN
;REPEAT TEST

```

;TEST FOR CORRECT OPERATION OF THE WAIT INSTRUCTION. A WAIT INSTRUCTION
;IS PERFORMED WHILE WAITING FOR A PUNCH INTERRUPT. WHEN THE INTERRUPT
;OCCURS, THE SERVICE ROUTINE CHANGES THE WAIT INSTRUCTION TO AN ERROR
;CALL AND THEN EXITS THE INTERRUPT WITH AN RTI. EXITING THE INTERRUPT
;SHOULD RETURN CONTROL TO THE INSTRUCTION FOLLOWING THE WAIT INSTRUCTION.
;IF CONTROL IS INSTEAD RETURNED TO THE SAME LOCATION WHERE THE WAIT
;INSTRUCTION WAS LOCATED AN ERROR CALL WILL OCCUR, INDICATING A FAILURE
;OF THE WAIT INSTRUCTION.

```

;TEST#
;NEXT TEST
;I COUNT
;SCOPE ENTRY
;WAIT 150 MSECS
;SET PUNCH INTERRUPT SERVICE
;TO AT41C
;MOVE WAIT INSTRUCTION TO AT41B
;LOAD PUNCH BUFFER (ENABLES PUNCH)
;ENABLE PUNCH INTERRUPTS
;SET PRIORITY 0.
;THIS LOCATION CAN BE EITHER
;A WAIT INSTRUCTION OR AN ERROR CALL.
;IF AN ERROR CALL IS EXECUTED, IT
;INDICATES A FAILURE OF THE WAIT INSTRUCTION.
;DISABLE PUNCH INTERRUPTS
;CHAIN
;REPEAT TEST
;MOVE ERROR CALL TO AT41B.
;EXIT INTERRUPT.

```

```

1401
1402
1403
1404 010000 000042
1405 010002 006052
1406 010004 000062
1407 010006 010014
1408 C10010 104400
1409 010012 000226
1410 010014 052777 000004 171074 AT42A: 150.
1411 010022 005077 171072 AT42A: BIT #BIT2,ATPS ;SET MAINTENANCE BIT
1412 010026 104400 CLR ATPB ;LOAD PUNCH BUFFER
1413 010030 000310 DELAY ;WAIT 200 MSECS
1414 010032 105777 171054 TSTB %TKS ;TEST READER DONE BIT
1415 010036 100401 BMI AT42B ;BRANCH IF READER DONE BIT SET.
1416 010040 104003 AT42E: ERROR ;ERROR. 200 MSECS AFTER PUNCH
1417 ;BUFFER LOAD WITH MAINTENANCE BIT
1418 ;SET THE READER DONE BIT WAS NOT SET
1419 010042 104012 AT42B: CHAIN ;CHAIN
1420 010044 000763 BR AT42A ;REPEAT TEST
1421 ;TESTS THAT DATA CAN BE RELIABLY
1422 ;TRANSFERRED WITH THE MAINTENANCE BIT
1423 ;SET. CAUSES PROGRAM 7 TO BE
1424 ;EXECUTED AS A SUBROUTINE IF THIS
1425 ;WAS A MONITOR LOAD OR IF THE
1426 ;MONITOR CELL IS NON-ZERO
1427 010046 000043 AT43: 43 ;TEST #
1428 010050 177777 -1 ;LAST TEST
1429 010052 001750 1000. ;ICOUNT
1430 010054 012402 LTA ;SCOPE ENTRY
1431 010056 005767 167760 TST MONITR ;IS THIS AUTOMATIC TESTING
1432 010062 001402 BEQ AT43B ;NO - EXIT
1433 010064 004767 002272 AT43A: JSR %7.PRG7 ;RUN PROGRAM 7
1434 010070 104012 AT43B: CHAIN ;CHAIN
1435 010072 004767 002304 JSR %7.LTA ;REPEAT TEST
1436 010076 000774 BR AT43B

```

```

1437
1438
1439 010100 012767 010132 171030
1440 010106 052767 000200 171036
1441 010114 012767 177600 174322
1442 010122 004767 175170
1443 010126 000167 172056
1444
1445 010132 000000
1446 010134 010234
1447 010136 104000
1448 010140 016656
1449 010142 012767 000111 171100
1450 010150 016767 171074 171106
1451 010156 005367 171102
1452 010162 001001
1453 010164 104012
1454 010166 016767 171072 171072
1455 010174 112700 000105
1456 010200 004767 174636
1457 010204 005367 171056
1458 010210 001371
1459 010212 112700 000015
1460 010216 004767 174620
1461 010222 012700 000012
1462 010226 004767 174610
1463 010232 000751
1464
1465 010234 000001
1466 010236 010300
1467 010240 104000
1468 010242 016707
1469 010244 012767 000016 171012
1470 010252 012767 016637 000014
1471 010260 104000
1472 010262 016631
1473 010264 005367 170774
1474 010270 001373
1475 010272 104007
1476 010274 000001
1477 010276 104012

:PRG1-PRINTER TESTS
PRG1:  MOV #CTO,KSTART ;SET ADDRESS IF 1ST ROUTINE.
      BIS #BIT7,PRGID ;BYPASS SCOPE AND ICNT.
      MOV #177600,STLMSK ;SET STALL LIMIT
      JSR %7,STBF ;SET UP BUFFER AREA.
      JMP SRSET ;GO GET STARTED.

:CARRIAGE RETURN TEST.
CTO:  0 ;TEST#
      CT1 ;NEXT TEST ADDRESS.
      TYPE ;TYPE TITLE.
      CRTST
      MOV #73.,RCNT
      MOV RCNT,CTRA ;RCNT TO CTRA
CTOA:  DEC CTRA ;DECREMENT CTRA
      BNE CTOB ;BRANCH IF NOT 0
      CHAIN ;O. CHAIN
CTOB:  MOV CTRA,CTRB ;SPACE COUNT TO CTRB.
CTOC:  MOVB #105,%D ;CHAR=E
      JSR %7,LSPCH ;SPACE.
      DEC CTRB ;DECREMENT CTRB.
      BNE CTOC ;BRANCH IF NOT DONE SPACING.
      MOVB #15,%D
      JSR %7,LSPCH ;CARRIAGE RETURN.
      MOV #12,%D ;LINE FEED
      JSR %7,LSPCH
      BR CTOA

:RIGHT MARGIN TEST
CT1:  1 ;TEST#
      CT2 ;NEXT TEST.
      TYPE ;TYPE TITLE
      RMTST
      MOV #14.,CTRA ;SET UP FOR 33/35
      MOV #RM33B,RMB
CT1A:  TYPE ;TYPE----I
      RM33A
      DEC CTRA ;DONE N TIMES.
      BNE CT1A ;BRANCH IF NOT N TIMES
      TYPE ;TYPE-I-.
RMB:  OPEN
      CHAIN ;CHAIN.

```

```

1478 ;CURSOR RIGHT TEST
1479 010300 000002 CT2: 2 ;TEST#
1480 010302 010452 CT3 ;NEXT TEST
1481 010304 104000 TYPE ;TYPE TITLE.
1482 010306 016735 SPTST
1483 010310 012767 000044 170746 MOV #36.,CTRA ;33/35 COUNT TO CTRA.
1484 010316 104000 CT2A: TYPE ;TYPE SPACE,\.
1485 010320 016653 SPTSTC
1486 010322 005367 170736 DEC CTRA ;DONE TIMES SET IN CTRA?
1487 010326 001373 BNE CT2A ;BRANCH IF NOT DONE
1488 010330 012767 000044 170726 MOV #36.,CTRA ;SET UP CTRA COUNT FOR 33/35
1489 010336 012767 000001 170722 CT2B: MOV #1,CTRB
1490 010344 016767 170716 CT2C: MOV CTRB,CTRC
1491 010352 112700 000015 MOVB #15,%D ;CARRIAGE RETURN.
1492 010356 004767 174460 JSR %7,LSPCH
1493 010362 004767 174454 JSR %7,LSPCH
1494 010366 005000 CT2D: CLR %D ;DUMMY CYCLE.
1495 010370 004767 174446 JSR %7,LSPCH ;NULL CHAR FOR FILLER
1496 010374 004767 174442 JSR %7,LSPCH ;TRANSMIT NULL CHAR
1497 010400 004767 174436 JSR %7,LSPCH ;TRANSMIT NULL CHAR
1498 010404 112700 000030 MOVB #30,%D ;CURSOR RIGHT
1499 010410 004767 174426 JSR %7,LSPCH ;SET IN CTCR.
1500 010414 005367 170650 DEC CTCR ;DONE SPACING.
1501 010420 001362 BNE CT2D ;BRANCH IF NOT DONE SPACING.
1502 010422 112700 000057 MOVB #',%D ;DONE. TYPE A "'".
1503 010426 004767 174410 JSR %7,LSPCH
1504 010432 005367 170626 DEC CTRA ;DONE 36 TIMES?
1505 010436 001001 BNE CT2E ;BRANCH IF NOT DONE.
1506 010440 104012 CHAIN ;DONE. CHAIN.
1507 010442 062767 000002 170616 CT2E: ADD #2,CTRB ;MODIFY CTRB FOR NEXT TRY.
1508 010450 000735 BR CT2C ;GO DO IT AGAIN.
1509 ;LINE FEED TEST
1510 010452 000003 CT3: 3 ;TEST #
1511 010454 010532 CT4 ;NEXT TEST
1512 010456 104000 TYPE ;TYPE TITLE
1513 010460 016763 LFTST
1514 010462 052767 040000 170462 BIS #BIT14,PRGID ;ALLOW STALLS.
1515 010470 012767 000110 170566 MOV #72.,CTRA ;SET 33/35 LINE FEED COUNT.
1516 010476 112700 000134 CT3A: MOVB #'\,%D ;TYPE "\"
1517 010502 004767 174334 JSR %7,LSPCH
1518 010506 112700 000012 MOVB #12,%D ;LINE FEED.
1519 010512 004767 174324 JSR %7,LSPCH
1520 010516 005367 170542 DEC CTRA ;DONE N TIMES?
1521 010522 001001 BNE CT3B ;BRANCH IF NOT DONE.
1522 010524 104012 CHAIN ;DONE. CHAIN
1523 010526 104002 CT3B: STALL ;STALL
1524 010530 000762 BR CT3A ;REPEAT

```


1525						:TAB TEST			
1526	010532	000004				CT4:	4		;TEST#
1527	010534	010750					CTS		;NEXT TEST.
1528	010536	012767	000011	000100			MOV	#9.,TBCNT	;SET TAB COUNT.
1529	010544	104014					CK35		;35?
1530	010546	104012					CHAIN		;NO.
1531	010550	004567	000044				JSR	%5,TPBM	;TYPE MARKERS
1532	010554	000007					7		
1533	010556	104000					TYPE		
1534	010560	016605					TBMRK+1		
1535	010562	012767	000007	170474	CT4A:		MOV	#7,CTRA	;LINE COUNT TO CTRA
1536	010570	005067	000052				CLR	SPCNT	;0 TO SPACE COUNT.
1537	010574	004767	000050		CT4B:		JSR	%7,TABP	;GO SPACE-TAB.
1538	010600	005267	000042				INC	SPCNT	;INCREMENT SPACE COUNT.
1539	010604	005367	170454				DEC	CTRA	;DONE ? LINES?
1540	010610	001371					BNE	CT4B	;BRANCH IF NOT DONE.
1541	010612	004767	174410				JSR	%7,TYPCLR	;CLEAR DOUBLE BUFFER
1542	010616	104012					CHAIN		;DONE: CHAIN.
1543	010620	012567	170440		TPBM:		MOV	(5)+,CTRA	;TYPE TEST TITLE.
1544	010624	104000					TYPE		
1545	010626	016570					TBTST		
1546	010630	104000			TPBMA:		TYPE		;TYPE MARKERS
1547	010632	016616					TBMRK1		
1548	010634	005367	170424				DEC	CTRA	
1549	010640	001373					BNE	TPBMA	
1550	010642	000205					RTS	%5	;EXIT.
1551	010644	000000			TBCNT:		OPEN		;TAB COUNT
1552	010646	000000			SPCNT:		OPEN		;SPACE COUNT
1553	010650	104000			TABP:		TYPE		;CRLF.
1554	010652	016627					CRLF		
1555	010654	016767	177764	170404			MOV	TBCNT,CTRB	;TAB COUNT TO CTRB
1556	010662	016767	177760	170400	TABPA:		MOV	SPCNT,CTRC	;SPACE COUNT TO CTCR
1557	010670	001407					BEQ	TABPC	;BRANCH IF SPACE COUNT IS 0.
1558	010672	112700	000040		TABPB:		MOVB	#40,%0	;SPACE
1559	010676	004767	174140				JSR	%7,LSPCH	
1560	010702	005367	170362				DEC	CTRC	;DECREMENT SPACE COUNT
1561	010706	001371					BNE	TABPB	;BRANCH IF NOT YET 0.
1562	010710	112700	000011		TABPC:		MOVB	#11,%0	;TAB
1563	010714	004767	174122				JSR	%7,LSPCH	
1564	010720	004767	174116				JSR	%7,LSPCH	;DUMMY CYCLE
1565	010724	004767	174112				JSR	%7,LSPCH	;DUMMY CYCLE.
1566	010730	112700	000057				MOVB	#',%0	;TYPE "'"
1567	010734	004767	174102				JSR	%7,LSPCH	
1568	010740	005367	170322				DEC	CTRB	;DECREMENT TAB COUNT.
1569	010744	001346					BNE	TABPA	;BRANCH IF NOT DONE TABBING.
1570	010746	000207					RTS	%7	;DONE. EXIT.

1571
1572
1573 010750 000005
1574 010752 010766
1575 010754 104000
1576 010756 017006
1577 010760 104016
1578 010762 016466
1579 010764 104012
1580
1581 010766 000006
1582 010770 011000
1583 010772 104016
1584 010774 016471
1585 010776 104012
1586
1587 011000 000007
1588 011002 011012
1589 011004 104016
1590 011006 016474
1591 011010 104012
1592
1593 011012 000010
1594 011014 011024
1595 011016 104016
1596 011020 016477
1597 011022 104012
1598
1599 011024 000011
1600 011026 011036
1601 011030 104016
1602 011032 016502
1603 011034 104012
1604
1605
1606 011036 000012
1607 011040 011050
1608 011042 104016
1609 011044 016505
1610 011046 104012
1611
1612 011050 000013
1613 011052 011062
1614 011054 104016
1615 011056 016510
1616 011060 104012

```

;TYPE LINE OF CHARACTERS ABC
CT5: 5 ;TEST #
      CT6 ;NEXT TEST
      TYPE ;TYPE "CHARACTER TESTS"
      CHRTST
      TYPLN3 ;TYPE LINE
      A
      CHAIN ;CHAIN
;TYPE LINE OF CHARACTERS DEF
CT6: 6 ;TEST #
      CT7 ;NEXT TEST
      TYPLN3 ;TYPE LINE
      D
      CHAIN ;CHAIN
;TYPE LINE OF CHARACTERS GHI
CT7: 7 ;TEST#
      CT10 ;NEXT TEST.
      TYPLN3 ;TYPE LINE
      G
      CHAIN ;CHAIN
;TYPE LINE OF CHARACTERS OF JKL
CT10: 10 ;TEST #
      CT11 ;NEXT TEST.
      TYPLN3 ;TYPELINE
      J
      CHAIN ;CHAIN
;TYPE LINE OF CHARACTERS MNO
CT11: 11 ;TEST #
      CT12 ;NEXT TEST
      TYPLN3 ;TYPE LINE
      M
      CHAIN ;CHAIN
;TYPE LINE OF CHARACTERS PQR
CT12: 12 ;TEST #
      CT13 ;NEXT TEST
      TYPLN3 ;TYPE LINE
      P
      CHAIN ;CHAIN
;TYPE LINE OF CHARACTERS STU
CT13: 13 ;TEST #
      CT14 ;NEXT TEST
      TYPLN3
      S
      CHAIN

```

1617			:TYPE LINE OF CHARACTERS VWX	
1618	011062	000014	CT14:	:TEST #
1619	011064	011074	14	:NEXT TEST
1620	011066	104016	CT15	:TYPE LINE
1621	011070	016513	TYPLN3	
1622	011072	104012	V	
1623			CHAIN	:CHAIN
1624			:TYPE LINE OF CHARACTERS YZD	
1625	011074	000015	CT15:	:TEST #
1626	011076	011106	15	:NEXT TEST
1627	011100	104016	CT16	:TYPE LINE
1628	011102	016516	TYPLN3	
1629	011104	104012	Y	
1630			CHAIN	:CHAIN
1631			:TYPE LINE OF CHARACTERS 123	
1632	011106	000016	CT16:	:TEST #
1633	011110	011120	16	:NEXT TEST
1634	011112	104016	CT17	:TYPE LINE
1635	011114	016521	TYPLN3	
1636	011116	104012	ONE	
1637			CHAIN	:CHAIN
1638			:TYPE LINE OF CHARACTERS 4 6	
1639	011120	000017	CT17:	:TEST #
1640	011122	011132	17	:NEXT TEST
1641	011124	104016	CT20	:TYPE LINE
1642	011126	016524	TYPLN3	
1643	011130	104012	FOUR	
1644			CHAIN	:CHAIN
1645			:TYPE LINE OF CHARACTERS 789	
1646	011132	000020	CT20:	:TEST #
1647	011134	011144	20	:NEXT TEST
1648	011136	104016	CT21	:TYPE LINE
1649	011140	016527	TYPLN3	
1650	011142	104012	SEVEN	
1651			CHAIN	:CHAIN
1652			:TYPE LINE OF CHARACTERS ! " #	
1653	011144	000021	CT21:	:TEST #
1654	011146	011156	21	:NEXT TEST
1655	011150	104016	CT22	:TYPE LINE
1656	011152	016533	TYPLN3	
1657	011154	104012	C4	
1658			CHAIN	:CHAIN
1659			:TYPE LINE OF CHARACTERS \$ % &	
1660	011156	000022	CT22:	:TEST #
1661	011160	011170	22	:NEXT TEST
1662	011162	104016	CT23	:TYPE LINE
1663	011164	016535	TYPLN3	
1664	011166	104012	C44	
1665			CHAIN	:CHAIN
1666			:TYPE LINE OF CHARACTERS * ()	
1667	011170	000023	CT23:	:TEST #
1668	011172	011202	23	:NEXT TEST
1669	011174	104016	CT24	:TYPE LINE
1670	011176	016540	TYPLN3	
1671	011200	104012	C47	
1672			CHAIN	:CHAIN.

1656									
1657									
1658	011202	000024							
1659	011204	011214							
1660	011206	104016							
1661	011210	016543							
1662	011212	104012							
1663									
1664	011214	000025							
1665	011216	011226							
1666	011220	104016							
1667	011222	016546							
1668	011224	104012							
1669									
1690	011226	000026							
1691	011230	011240							
1692	011232	104016							
1693	011234	016551							
1694	011236	104012							
1695									
1696	011240	000027							
1697	011242	011252							
1698	011244	040016							
1699	011246	016554							
1700	011250	104012							
1701									
1702	011252	000030							
1703	011254	011264							
1704	011256	104016							
1705	011260	016557							
1706	011262	104012							
1707									
1708	011264	000031							
1709	011266	011276							
1710	011270	104016							
1711	011272	016562							
1712	011274	104012							
1713									
1714	011276	000032							
1715	011300	011334							
1716	011302	004767	174072						
1717	011306	042767	040000	167636					
1718	011314	004767	173656						
1719	011320	052767	040000	167624					
1720	011326	004767	173644						
1721	011332	104012							

```

:TYPE LINE OF CHARACTERS *+.
CT24: 24 :TEST #
      CT25 :NEXT TEST
      TYPLN3 :TYPE LINE
      CS2
      CHAIN ;CHAIN
:TYPE LINE OF CHARACTERS -./
CT25: 25 :TEST #
      CT26 :NEXT TEST
      TYPLN3 :TYPE LINE
      CS5
      CHAIN ;CHAIN
:TYPE LINE OF CHARACTERS ::<
CT26: 26 :TEST #
      CT27 :NEXT TEST
      TYPLN3 :TYPE LINE
      CT2
      CHAIN ;CHAIN
:TYPE LINE OF CHARACTERS =>?
CT27: 27 :TEST #
      CT30 :NEXT TEST
      TYPLN3 :TYPE LINE
      CT5
      CHAIN ;CHAIN.
:TYPE LINE OF CHARACTERS @|\
CT30: 30 :TEST #
      CT31 :NEXT TEST
      TYPLN3 :TYPE LINE
      C100
      CHAIN ;CHAIN
:TYPE LINE OF CHARACTERS |^AND LEFT ARROW
CT31: 31 :TEST #
      CT32 :NEXT TEST
      TYPLN3 :TYPE LINE
      C125
      CHAIN ;CHAIN
:TYPE 2 LINES OF ALL CHARACTERS. FIRST LINE FULL SPEED. SECOND LINE WITH STALLS.
CT32: 32 :TEST #
      CT33 :NEXT TEST.
      JSR %7,FBALL :FILL BUFFER WITH ALL CHARACTERS.
      BIC #BIT14,PRGID :CLEAR STALL BIT IN PRGID
      JSR %7,TYPLN :TYPE LINE.
      BIS #BIT14,PRGID :SET STALL BIT IN PRGID
      JSR %7,TYPLN :TYPE LINE.
      CHAIN ;CHAIN.

```

```

1712
1713
1714 011334 000033
1715 011336 011416
1716 011340 104013
1717 011342 104012
1718 011344 104000
1719 011346 017032
1720 011350 004767 174064
1721 011354 012767 000006 167702
1722 011362 042767 040000 167562
1723 011370 004767 173602
1724 011374 052767 040000 167550
1725 011402 004767 173570
1726 011406 005367 167652
1727 011412 001363
1728 011414 104012
1729
1730 011416 000034
1731 011420 011500
1732 011422 104014
1733 011424 104012
1734 011426 104000
1735 011430 017032
1736 011432 004767 174042
1737 011436 012767 000006 167620
1738 011444 042767 040000 167500
1739 011452 004767 173520
1740 011456 052767 040000 167466
1741 011464 004767 173506
1742 011470 005367 167570
1743 011474 001363
1744 011476 104012
1745
1746
1747
1748 011500 000035
1749 011502 177777
1750 011504 005767 166332
1751 011510 001404
1752 011512 004767 001116
1753 011516 004767 173504
1754 011522 104012

```

;TYPE 12 LINES OF ASR33 WORST CASE PATTERN. ALTERNATE LINES WITH STALLS.

```

CT33: 33 ;TEST #
      CT34 ;NEXT TEST
      CK33 ;33?
      CHAIN ;NO. BYPASS TEST.
      TYPE ;TYPE "WORST CASE PATTERN TEST"
      WCPYST
      JSR %7,FW336 ;PATTERN TO BUFFER.
      MOV #6,CTRA ;SET COUNT TO 6
      CT33A: BIC #BIT14,PRGID ;CLEAR STALL BIT IN PRGID.
      JSR %7,TYPLN ;TYPE LINE
      BIS #BIT14,PRGID ;SET STALL BIT IN PRGID.
      JSR %7,TYPLN ;TYPE LINE.
      DEC CTRA ;DONE 6 TIMES?
      BNE CT33A ;BRANCH IF NOT 6 TIMES YET.
      CHAIN ;DONE. CHAIN.

```

;TYPE 12 LINES OF ASR35 WORST CASE PATTERN. ALTERNATE LINES WITH STALLS.

```

CT34: 34 ;TEST #
      CT35 ;NEXT TEST.
      CK35 ;35?
      CHAIN ;NO. BYPASS TEST.
      TYPE ;TYPE "WORST CASE PATTERN TEST"
      WCPYST
      JSR %7,FW356 ;PATTERN TO BUFFER.
      MOV #6,CTRA ;SET COUNT TO 6.
      CT34A: BIC #BIT14,PRGID ;CLEAR STALL BIT IN PRGID.
      JSR %7,TYPLN ;TYPE LINE.
      BIS #BIT14,PRGID ;SET STALL BIT IN PRGID.
      JSR %7,TYPLN ;TYPE LINE
      DEC CTRA ;DONE 6 TIMES?
      BNE CT34A ;BRANCH IF NOT 6 TIMES YET.
      CHAIN ;CHAIN.

```

```

;RUN PROGRAM 11 AS A SUBROUTINE TO EXERCISE
;CURSOR ADDRESSING FUNCTIONS IF THIS WAS A MONITOR
;LOAD OR IF CELL MONITOR IS NON ZERO.

```

```

CT35: 35 ;TEST #
      -1 ;LAST TEST
      TST MONTR ;IS THIS AN AUTOMATIC TEST?
      BEQ CT35A ;NO - EXIT
      JSR %7,PRG11 ;RUN PROG. 11 TEST
      JSR %7,TYPCLR ;CLEAR DOUBLE BUFFER
      CT35A: CHAIN ;DONE CHAIN

```

```

1755
1756
1757 011524 012767 011550 167404 ;PRG2-KEYBOARD TEST
1758 011532 052767 000200 167412 PRG2:  MOV #ETO,KSTART
1759 011540 104000 TYPE #BIT7,PRGID
1760 011542 017066 KMSG1
1761 011544 000167 170440 JMP SRSET
1762 ;TEST THAT PRESSING KEY SETS DONE FLAG.
1763 011550 000000 ETO: 0 ;TEST #
1764 011552 011654 ET1 ;NEXT TEST.
1765 011554 012767 000005 167502 MOV #5,CTRA
1766 011562 104006 ETOA: STRDRV
1767 011564 011620 ETOB
1768 011566 104000 TYPE ;TYPE "PRESS A KEY WITHIN 10 SECS."
1769 011570 017104 KMSG2
1770 011572 052777 000100 167312 BIS #BIT6,@TKS ;ENABLE KYBD INTERRUPT.
1771 011600 005067 166172 CLR PSW
1772 011604 104400 DELAY ;WAIT 10 SECONDS
1773 011606 023420 10000.
1774 011610 104000 TYPE ;TYPE "NO KEYBOARD REQUEST."
1775 011612 017306 KMSG6
1776 011614 104010 EHALT ;HALT.
1777 011616 000411 BR ETOCA
1778 011620 105777 167266 ETOB: TSTB @TKS ;TEST FOR DONE BIT ON
1779 011624 100403 BMI ETOC ;BRANCH IF DONE BIT SET.
1780 011626 104000 TYPE ;DONE BIT NOT SET. TYPE:FALSE KEY-
1781 011630 017334 KMSG7 ;BOARD OR READER INTERRUPT.
1782 011632 104010 EHALT ;HALT
1783 011634 012716 011642 ETOC: MOV #ETOCA,@%6
1784 011640 000002 RTI ;EXIT INTERRUPT.
1785 011642 104011 ETOCA: SRESET
1786 011644 005367 167414 DEC CTRA ;DONE 5 TIMES?
1787 011650 001344 BNE ETOA ;BRANCH IF NOT DONE.
1788 011652 104012 CHAIN ;CHAIN
1789 ;ECHO TEST. KEYED CHARACTER IS TYPED. RUBOUT ENDS ROUTINE.
1790 011654 000001 ET1: 1 ;TEST #
1791 011656 011734 ET2 ;NEXT TEST.
1792 011660 104000 TYPE ;TYPE TITLE AND INSTRUCTIONS.
1793 011662 017144 KMSG3
1794 011664 105777 167222 ET1A: TSTB @TKS ;WAIT FOR DONE FLAG
1795 011670 100375 BPL -4
1796 011672 117767 167216 167352 MOVB @TKB,CRBUF ;MOVE KYBD CHAR TO CRBUF.
1797 011700 116777 167346 167212 MOVB CRBUF,@TPB ;ECHO CHAR READ.
1798 011706 105777 167204 TSTB @TPS ;WAIT FOR PRINTER DONE.
1799 011712 100375 BPL -4
1800 011714 042767 000200 167330 BIC #BIT7,CRBUF ;CLEAR BIT 7 FROM CRBUF.
1801 011722 122767 000177 167322 CMPB #177,CRBUF ;COMPARE CRBUF TO RUBOUT (177)
1802 011730 001355 BNE ET1A ;BRANCH IF NOT RUBOUT (177)
1803 011732 104012 CHAIN ;CHAIN

```

```

1804
1805
1806
1807 011734 000002
1808 011736 177777
1809 011740 104001
1810 011742 017246
1811 011744 017157
1812 011746 177777
1813 011750 005067 167276
1814 011754 105777 167132
1815 011760 100375
1816 011762 117767 167126 167262
1817 011770 004567 172676
1818 011774 001252
1819 011776 017300
1820 012000 104000
1821 012002 017276
1822 012004 042767 000200 167240
1823 012012 022767 000177 167232
1824 012020 001355
1825 012022 104012

```

```

;OCTAL EQUIVALENT TEST. THE OCTAL EQUIVALENT OF ANY CHARACTER KEYED
;IS PRINTED. RUBOUT ENDS ROUTINE.

```

```

ET2: 2 ;TEST #
-1 ;LAST TEST
TYPES ;TYPE TITLE AND INSTRUCTIONS.
KMSG4
KMSG3A
-1
CLR CRBUF
ET2A: TSTB @TKS ;WAIT FOR DONE FLAG.
BPL -4
MOVB @TKB,CRBUF ;CHARACTER TO CRBUF
JSR %5,ACNV4 ;CONVERT CHAR IN CRBUF TO
;PRINTABLE OCTAL
TYPE ;TYPE OCTAL EQUIVALENT
KMSG5
BIC #BIT7,CRBUF ;CLEAR BIT 7 FROM CRBUF
CMP #177,CRBUF ;TEST FOR RUBOUT CHARACTER.
BNE ET2A ;BRANCH IF NOT RUBOUT (177).
CHAIN ;CHAIN.

```

```

1826
1827
1828
1829 012024 004767 173266
1830 012030 104000
1831 012032 017362
1832 012034 052767 040000 167110
1833 012042 012767 177600 172374
1834 012050 012703 017443
1835 012054 104000
1836 012056 017410
1837 012060 012767 000006 167176
1838 012066 004767 000060
1839 012072 005367 167166
1840 012076 001373
1841 012100 042767 000200 167144
1842 012106 122767 000177 167136
1843 012114 001003
1844 012116 042767 040000 167026
1845 012124 004567 172654
1846 012130 017443
1847 012132 017450
1848 012134 000103
1849 012136 004767 173034
1850 012142 005767 165422
1851 012146 100732
1852 012150 000772
1853 012152 105777 166734
1854 012156 100375
1855 012160 117767 166730 167064
1856 012166 116713 167060
1857 012172 142723 000200
1858 012176 116700 167050
1859 012202 004767 172634
1860 012206 000207

;PRG3-PRINTER EXERCISER. KEYBOARD CONTROLLED.
;TYPES LINES WITH ANY 5 CHARACTERS. STALLS OR FULL SPEED.
PRG3: JSR %7,STBF ;SET UP BUFFER.
      TYPE ;TYPE TITLE
      P7MG1
HTA: BIS #BIT14,PRGID ;SET STALL BIT IN PRGID.
      MOV #177600,STLMSK ;SET STALL MASK
      MOV #BLOCK1,%3
      TYPE ;TYPE "TYPE IN DATA".
      P7MG2
HTB: MOV #6,CTRA ;CHAR COUNT TO CTRA.
      JSR %7,GKBCR ;GET AND STORE KYBD CHARACTER.
      DEC CTRA ;GOT 6 CHARACTERS?
      BNE HTB ;BRANCH IF NOT 6 CHARS YET.
      BIC #BIT7,CRBUF
      CMPB #177,CRBUF ;CHECK 6TH CHAR FOR RUBOUT.
      BNE HTC ;BRANCH IF NOT A RUBOUT.
      BIC #BIT14,PRGID ;RUBOUT. CLEAR STALL BIT IN PRGID.
HTC: JSR %5,BMOVE ;FILL 72 CHAR LINE.
      BLOCK1
      BLOCK1+5
      67.
HTD: JSR %7,TYPLN ;TYPE LINE.
      TST SR ;CHANGE DATA? (SR15=1).
      BMI HTA ;YES. GO CHANGE DATA
      BR HTD ;NO CONTINUE WITH SAME DATA.
GKBCR: TSTB @TKS ;WAIT FOR DONE FLAG.
      BPL .-4
      MOVB @TKB,CRBUF ;CHARACTER TO CRBUF.
      MOVB CRBUF,(3) ;CHARACTER TO LINE BUFFER.
      BICB #200,(3)+
      MOVB CRBUF,%0
      JSR %7,LSPCH ;ECHO CHARACTER.
      RTS %7

```



```

1861
1862
1863
1864
1865
1866 012210 104005
1867 012212 004767 000036
1868 012216 000775
1869
1870
1871
1872
1873
1874
1875
1876
1877 012220 104005
1878 012222 004767 000020
1879 012226 017700 166662
1880 012232 000005
1881 012234 000005
1882 012236 000005
1883 012240 000005
1884 012242 000005
1885 012244 000766
1886
1887 012246 052777 000004 166642
1888 012254 116767 165310 000022
1889 012262 005767 000016
1890 012266 001002
1891 012270 005267 000010
1892 012274 116777 165271 166616
1893 012302 104400
1894 012304 000000
1895 012306 000207

```

```

:
: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.
PRG4: CHALT ;HALT TO SET SR.
ITA: JSR %7,C1112 ;GO OUTPUT CHARACTER SET IN LEFT
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 RO.
PRG5: CHALT ;HALT TO SET SR.
JTA: JSR %7,C1112M ;GO OUTPUT CHARACTER FROM SR LEFT AND
MOV %TKB,%0 ;STALL PER SR RIGHT. (TKB) TO RO.
RESET ;"FIX" (TKB) IN DATA LIGHTS.
RESET
RESET
RESET
RESET
BR JTA ;REPEAT.
:
C1112M: BIS #4,%TPS ;SET MAINTENANCE MODE (PUNCH).
C1112: MOVB SR,XTY ;STALL COUNT TO XTY.
TST XTY ;DISREGARD 0 DELAY.
BNE C1112A
INC XTY
C1112A: MOVB SR+1,%TPB ;LOAD PUNCH BUFFER.
DELAY ;DELAY (APPROXIMATELY) THE NUMBER OF
XTY: OPEN ;MSECS. SPECIFIED AT SR RIGHT
RTS %7 ;EXIT

```

1896
1897
1898
1899
1900
1901
1902
1903
1904
1905 012310 104005
1906 012312 052777 000004 166576
1907 012320 105777 166572
1908 012324 100375
1909 012326 116767 165237 166717
1910 012334 116777 166713 166556
1911 012342 105777 166544
1912 012346 100375
1913 012350 117767 166540 166674
1914 012356 104004
1915 012360 000754
1916
1917
1918
1919
1920 012362 004767 172104
1921 012366 012767 177600 172050
1922 012374 052767 040000 166550
1923 012402 052777 000004 166506
1924 012410 032767 000400 165152
1925 012416 001001
1926 012420 104002
1927 012422 105777 166470
1928 012426 100375
1929 012430 004767 172142
1930 012434 110167 166613
1931 012440 110177 166454
1932 012444 105777 166442
1933 012450 100375
1934 012452 117767 166436 166572
1935 012460 104004
1936 012462 005767 165354
1937 012466 001401
1938 012470 000207
1939 012472 000743
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950 012474 012767 177736 000126
1951 012502 016767 000122 000122

;PRG6-MAINTENANCE MODE SINGLE CHARACTER DATA TEST.
;WITH MAINTENANCE MODE SET, OUTPUTS ONTO PUNCH BUFFER AND BACK ONTO
;READER BUFFER THE CHARACTER SET IN SR LEFT. THE CHARACTER IN THE
;READER BUFFER IS COMPARED TO THE CHARACTER IN SR LEFT. IF THE 2 CHARACTERS
;DISAGREE THE PROGRAM HALTS. THE DATA LIGHTS WILL THEN CONTAIN:

;LEFT HALF: THE EXPECTED CHARACTER (SR LEFT).
;RIGHT HALF: THE CHARACTER IN THE READER BUFFER.
PRG6: CHALT ;HALT TO SET SR.
KTA: BIS #4,ATPS ;SET MAINTENANCE MODE.
KTB: TSTB ATPS ;WAIT FOR READY.
BPL -4
MOVB SR+1,CRBUF+1 ;S/B CHAR TO CRBUF+1.
MOVB CRBUF+1,ATPB ;OUTPUT CHARACTER.
TSTB ATKS ;WAIT FOR READER DONE FLAG.
BPL -4
MOVB ATKB,CRBUF ;CHAR READ TO CRBUF.
DATCHK ;GO CHECK AGAINST S/B CHAR.
BR KTA ;REPEAT.

;PRG7-MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST.
;PERFORMS SAME OPERATION AS PRG6, EXCEPT THAT SPECIAL BINARY COUNT
;PATTERN IS USED.

PRG7: JSR %7,INBIN ;INITIALIZE BINARY COUNT
MOV #177600,STLMSK ;SET STALL LIMIT
BIS #BIT14,PRGID ;ALLOW STALLS
LTA: BIS #4,ATPS ;SET MAINTENANCE MODE.
BIT #BIT8,SR ;CHECK STALL SWITCH
BNE LTB ;BRANCH IF NO STALL WANTED
STALL ;STALL
LTB: TSTB ATPS ;WAIT FOR READY.
BPL -4
JSR %7,GTBINP ;GET BIN CHARACTER.
MOVB %1,CRBUF+1 ;MOVE TO S/B CHAR.
MOVB %1,ATPB ;OUTPUT BIN CHARACTER.
TSTB ATKS ;WAIT FOR READER DONE.
BPL -4
MOVB ATKB,CRBUF ;CHAR IN READ BUFFER TO CRBUF.
DATCHK ;GO CHECK AGAINST S/B CHAR.
TST MONITR ;MONITOR LOAD?
BEQ LTC ;NO - CONTINUE
RTS %7 ;YES - RETURN
LTC: BR LTA ;CONTINUE.

;PRG10 ROLE UP TEST
;THE FUNCTION OF THIS TEST IS TO TEST THE ROLL-UP CAPABILITY
;OF THE VT05
;TO DO THIS A LINE OF A CHARACTER AND IT'S COMPLEMENT FOLLOWED
;BY A LINE OF THE COMPLEMENT AND THE CHARACTER IS TRANSMITTED
;THIS SCHEME IS CONTINUED UNTIL SWITCH 15 IS RAISED
;THE CHARACTER SHOULD NOT BE CHANGED UNTIL THE SCREEN HAS BEEN
;COMPLETELY FILLED

PRG10: MOV #-42,TCHAR ;INIT TEMP CHAR
RENIT: MOV TCHAR,CHAR ;COMPLEMENT OF "!"

1952	012510	012767	177670	000110	PRG10C: MOV	#-72.,CNT	:72 CHAR/LINE
1953	012516	005167	000110		PRG10D: COM	CHAR	
1954	012522	016700	000104		PRG10A: MOV	CHAR,%0	:LOAD "!"
1955	012526	004767	172310		JSR	%7,LSPCH	:PUNCH "!"
1956	012532	005167	000074		COM	CHAR	:COMPLEMENT TO "!"
1957	012536	016700	000070		MOV	CHAR,%0	:LOAD "!"
1958	012542	004767	172274		JSR	%7,LSPCH	:PUNCH "!"
1959	012546	005167	000060		COM	CHAR	
1960	012552	062767	000002	000046	ADD	#2,CNT	:END OF LINE?
1961	012560	001360			BNE	PRG10A	:NO
1962	012562	012700	000015		MOV	#15,%0	:CR
1963	012566	004767	172250		JSR	%7,LSPCH	
1964	012572	012700	000012		MOV	#12,%0	:LF
1965	012576	004767	172240		JSR	%7,LSPCH	
1966	012602	005767	164762		TST	SR	:NEXT CHAR
1967	012606	100340			BPL	PRG10C	:NO
1968	012610	005367	000014		DEC	TCHAR	:YES CHANGE TCHAR
1969	012614	022767	177677	000006	CMP	#177677,TCHAR	:CHAR STRING COMPLETE
1970	012622	001724			BEQ	PRG10	
1971	012624	000726			BR	RENIT	
1972	012626	000000			CNT:	0	
1973	012630	177735			TCHAR:	-42	
1974	012632	000041			CHAR:	41	
1975							
1976					:PRG11-CURSOR ADDRESS TEST		
1977					:RANDOM NUMBERS ARE GENERATED AND USED AS "X" AND "Y" COORDINATES		
1978					:ADDRESSING A 1440 CHARACTER PRINTOUT.		
1979					:VERIFICATION OF DISPLAY IS PERFORMED VISUALLY BY THE USER		
1980					:THIS PROGRAM MAY BE RERUN AFTER COMPLETION, BY DEPRESSING THE CONT. KEY		
1981							
1982	012634	005067	000442		PRG11: CLR	OVRAL	
1983	012640	000167	000262		JMP	SRT	
1984	012644	012700	000035		PGO: MOV	#35,%0	:LOAD "CURSOR HOME"
1985	012650	004767	172166		JSR	%7,LSPCH	:PRINT (MOVE) CURSOR "HOME"
1986	012654	012700	000037		MOV	#37,%0	:LOAD "PAGE ERASE"
1987	012660	004767	172156		JSR	%7,LSPCH	:ERASE CRT FACE
1988	012664	016767	000414	000406	GENER: MOV	RTWY1,SET	:SETUP PRINTOUT
1989	012672	004767	171014		JSR	%7,RNGEN	:GENERATE RANDOM NUMBER
1990	012676	005100			COM	%0	
1991	012700	042700	177700		BIC	#177700,%0	:RANDOM NO. MUST BE TWO DIGITS
1992	012704	020027	000037		CMP	%0,#37	:NO. MUST BE LESS THAN 40
1993	012710	101765			BLOS	GENER	:LOWER, REGENERATION
1994	012712	020067	000352		CMP	%0,HYCOR	:NO. MUST NOT BE GREATER THAN 63
1995	012716	101362			BHI	GENER	:GREATER, REGENERATION
1996	012720	010067	000346		MOV	%0,YADDS	:STORE RANDOM Y COORDINATE
1997	012724	010001			MOV	%0,%1	:STORE Y COORDINATE IN HI BYTE OF BUFFER
1998	012726	166701	000334		SUB	LYCOR,%1	:MINIMUM X,Y COORDINATE
1999	012732	001405			BEQ	GENRX	:RESULT, MINIMUM Y COORDINATE
2000	012734	062767	000110	000336	RATZ: ADD	#110,SET	:SETUP MINIMUM Y LOCATION FOR PRINTOUT
2001	012742	005301			DEC	%1	
2002	012744	001373			BNE	RATZ	:Y COORDINATE IS SET
2003	012746	004767	170740		GENERX: JSR	%7,RNGEN	:GENERATE RANDOM NUMBER
2004	012752	005100			COM	%0	
2005	012754	042700	177600		BIC	#177600,%0	:RANDOM NO. MAY BE LESS THAN 200
2006	012760	020027	000037		CMP	%0,#37	:NO. MUST NOT BE LESS THAN 40
2007	012764	101770			BLOS	GENERX	:LOWER, REGENERATION

2008	012766	020067	000302		CMP	%0,HXCOR	;NO. MUST NOT BE GREATER THAN 147	
2009	012772	101365			BHI	GENRX	;GREATER, REGENERATION	
2010	012774	010067	000276		MOV	%0,XADD5	;STORE RANDOM X COORDINATE	
2011	013000	166700	000262		SUB	LYCOR,%0	;SETUP MINIMUM LOCATION	
2012	013004	060067	000270		ADD	%0,SET	;SETUP X COOR. FOR PNTOUT.	
2013	013010	016701	000264		MOV	SET,%1	;SETUP CHECK	
2014	013014	105711			TSTB	(1)	;HAS CURRENT CHAR. ALREADY BEEN USED?	
2015	013016	100502			BMI	TROB	;YES, INCREMENT RANDOM PATTERNS	
2016	013020	012700	000016		MOV	#16,%0	;LOAD CURSOR ADDRESSING CODE	
2017	013024	004767	172012		JSR	%7,LSPCH	;PRINT CAD	
2018	013030	016700	000236		MOV	YADD5,%0	;LOAD Y COORDINATE	
2019	013034	004767	172002		JSR	%7,LSPCH	;PRINT (MOVE) CURSOR TO Y COORDINATE	
2020	013040	016700	000232		MOV	XADD5,%0	;LOAD X COORDINATE	
2021	013044	004767	171772		JSR	%7,LSPCH	;PRINT (MOVE) CURSOR TO X COORDINATE	
2022	013050	111100			MOVB	(1),%0	;LOAD CHARACTER TO BE PRINTED	
2023	013052	004767	171764		JSR	%7,LSPCH	;PRINT CHARACTER	
2024	013056	152711	000200		BISB	#200,(1)	;INDICATE USE OF CURSOR POSITION	
2025	013062	005367	000214		DEC	OVRAL	;MAXIMUM NO. OF COORDINATES	
2026	013066	001402			BEQ	RESRT	;IS CRT COMPLETE?	
2027	013070	000167	177570		JMP	GENER	;NO, GENERATE ANOTHER NUMBER	
2028	013074	005767	164742	RESRT:	TST	MONITR	;AUTOMATIC TESTING DESIRED?	
2029	013100	001401			BEQ	RESRT1	;NO - HALT	
2030	013102	000207			RTS	%7	;YES - RETURN	
2031	013104	000000		RESRT1:	HALT		;WAIT FOR CONTINUE	
2032	013106	012767	001233	170644	MOV	#1233,RP1	;SETUP RANDOM GENERATOR	
2033	013114	012767	005622	170640	MOV	#5622,RP2		
2034	013122	000167	177506		JMP	PRG11		
2035	013126	016703	000152		SRT:	MOV	RTWY1,%3	;SETUP BUFFER FOR REFRESHING
2036	013132	142723	000200		SRT1:	BICB	#200,(3)+	;REFRESH BUFFER
2037	013136	005267	000140		INC	OVRAL	;MAXIMUM NO. OF COORDINATES	
2038	013142	026727	000134	002640	CMP	OVRAL,#2640	;RESET MAX. NO OF CHARACTERS	
2039	013150	001402			BEQ	WT	;ONCE RESET, WAIT FOR VISUAL INSP.	
2040	013152	000167	177754		JMP	SRT1	;IF NOT RESET, CONTINUE	
2041	013156	012767	000010	000074	WT:	MOV	#10,TWW	;RESET COUNTERS
2042	013164	012767	000001	000070	MOV	#1,STEP		
2043	013172	012767	000001	000064	MOV	#1,STEP1		
2044	013200	000167	177440		JMP	PG0	;RESTART	
2045	013204	016767	000052	170546	NUM:	MOV	STEP,RP1	;USED BY RANDOM NO. GENERATOR
2046	013212	016767	000046	170542	MOV	STEP1,RP2		
2047	013220	000167	177440		JMP	GENER		
2048	013224	005367	000030		TROB:	DEC	TWW	;NOS. BEING GENERATED HAVE BEEN USED
2049	013230	001402			BEQ	SAD		
2050	013232	000167	177426		JMP	GENER		
2051	013236	012767	000010	000014	SAD:	MOV	#10,TWW	;CHANGE RANDOM PATTERNS
2052	013244	005267	000012		INC	STEP		
2053	013250	005267	000010		INC	STEP1		
2054	013254	000167	177724		JMP	NUM		
2055								
2056	013260	000010			TWW:	10		
2057	013262	000001			STEP:	1		
2058	013264	000001			STEP1:	1		
2059	013266	000040			LYCOR:	40		
2060	013270	000063			HXCOR:	63		
2061	013272	000000			YADD5:	OPEN		
2062	013274	000147			HXCOR:	147		
2063	013276	000000			XADD5:	OPEN		

2064	013300	000000		
2065	013302	000000		
2066	013304	013320		
2067	013306	004777	164530	
2068	013212	000240		
2069	013314	000240		
2070	013316	000240		
2071	013320	042074	043511	052111
2072	013326	046101	042452	052521
2073	013334	050111	042515	052116
2074	013342	041452	051117	047520
2075	013350	040522	044524	047117
2076	013356	021076	021042	021042
2077	013364	021042	021042	021042
2078	013372	021042	021042	021042
2079	013400	021042	021042	021042
2080	013406	021042	021042	021042
2081	013414	021042	021042	021042
2082	013422	021042	021042	021042
2083	013430	042074	043511	052111
2084	013436	046101	042452	052521
2085	013444	050111	042515	052116
2086	013452	041452	051117	047520
2087	013460	040522	044524	047117
2088	013466	021476	021443	021443
2089	013474	021443	021443	021443
2090	013502	021443	021443	021443
2091	013510	021443	021443	021443
2092	013516	021443	021443	021443
2093	013524	021443	021443	021443
2094	013532	021443	021443	021443
2095	013540	042074	043511	052111
2096	013546	046101	042452	052521
2097	013554	050111	042515	052116
2098	013562	041452	051117	047520
2099	013570	040522	044524	047117
2100	013576	022076	022044	022044
2101	013604	022044	022044	022044
2102	013612	022044	022044	022044
2103	013620	022044	022044	022044
2104	013626	022044	022044	022044
2105	013634	022044	022044	022044
2106	013642	022044	022044	022044
2107	013650	042074	043511	052111
2108	013656	046101	042452	052521
2109	013664	050111	042515	052116
2110	013672	041452	051117	047520
2111	013700	040522	044524	047117
2112	013706	022476	022445	022445
2113	013714	022445	022445	022445
2114	013722	022445	022445	022445
2115	013730	022445	022445	022445
2116	013736	022445	022445	022445
2117	013744	022445	022445	022445
2118	013752	022445	022445	022445
2119	013760	042074	043511	052111

SET: OPEN
CVRAL: OPEN
RTWY1: RTWY
LOGICAL: JSR %7,AMONITR
NOP
NOP
NOP
RTWY: .ASCII '<DIGITAL*EQUIPMENT*CORPORATION>'

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>'

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>'

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>'

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>'

2120	013766	046101	042452	052521
2121	013774	050111	042515	052116
2122	014002	041452	051117	047520
2123	014010	040522	044524	047117
2124	014016	023076	023046	023046
2125	014024	023046	023046	023046
2126	014032	023046	023046	023046
2127	014040	023046	023046	023046
2128	014046	023046	023046	023046
2129	014054	023046	023046	023046
2130	014062	023046	023046	023046
2131	014070	042074	043511	052111
2132	014076	046101	042452	052521
2133	014104	050111	042515	052116
2134	014112	041452	051117	047520
2135	014120	040522	044524	047117
2136	014126	024076	024050	024050
2137	014134	024050	024050	024050
2138	014142	024050	024050	024050
2139	014150	024050	024050	024050
2140	014156	024050	024050	024050
2141	014164	024050	024050	024050
2142	014172	024050	024050	024050
2143	014200	042074	043511	052111
2144	014206	046101	042452	052521
2145	014214	050111	042515	052116
2146	014222	041452	051117	047520
2147	014230	040522	044524	047117
2148	014236	024476	024451	024451
2149	014244	024451	024451	024451
2150	014252	024451	024451	024451
2151	014260	024451	024451	024451
2152	014266	024451	024451	024451
2153	014274	024451	024451	024451
2154	014302	024451	024451	024451
2155	014310	042074	043511	052111
2156	014316	046101	042452	052521
2157	014324	050111	042515	052116
2158	014332	041452	051117	047520
2159	014340	040522	044524	047117
2160	014346	025076	025052	025052
2161	014354	025052	025052	025052
2162	014362	025052	025052	025052
2163	014370	025052	025052	025052
2164	014376	025052	025052	025052
2165	014404	025052	025052	025052
2166	014412	025052	025052	025052
2167	014420	042074	043511	052111
2168	014426	046101	042452	052521
2169	014434	050111	042515	052116
2170	014442	041452	051117	047520
2171	014450	040522	044524	047117
2172	014456	025476	025453	025453
2173	014464	025453	025453	025453
2174	014472	025453	025453	025453
2175	014500	025453	025453	025453

.ASCII ' <DIGITAL*EQUIPMENT*CORPORATION>((

.ASCII ' <DIGITAL*EQUIPMENT*CORPORATION>))

.ASCII ' <DIGITAL*EQUIPMENT*CORPORATION>*****

.ASCII ' <DIGITAL*EQUIPMENT*CORPORATION>+++++

```

2176 014506 025453 025453 025453
2177 014514 025453 025453 025453
2178 014522 025453 025453 025453
2179 014530 042074 043511 052111
2180 014536 046101 042452 052521
2181 014544 050111 042515 052116
2182 014552 041452 051117 047520
2183 014560 040522 044524 047117
2184 014566 036476 036475 036475
2185 014574 036475 036475 036475
2186 014602 036475 036475 036475
2187 014610 036475 036475 036475
2188 014616 036475 036475 036475
2189 014624 036475 036475 036475
2190 014632 036475 036475 036475
2191 014640 042074 043511 052111
2192 014646 046101 042452 052521
2193 014654 050111 042515 052116
2194 014662 041452 051117 047520
2195 014670 040522 044524 047117
2196 014676 027076 027056 027056
2197 014704 027056 027056 027056
2198 014712 027056 027056 027056
2199 014720 027056 027056 027056
2200 014726 027056 027056 027056
2201 014734 027056 027056 027056
2202 014742 027056 027056 027056
2203 014750 042074 043511 052111
2204 014756 046101 042452 052521
2205 014764 050111 042515 052116
2206 014772 041452 051117 047520
2207 015000 040522 044524 047117
2208 015006 027476 027457 027457
2209 015014 027457 027457 027457
2210 015022 027457 027457 027457
2211 015030 027457 027457 027457
2212 015036 027457 027457 027457
2213 015044 027457 027457 027457
2214 015052 027457 027457 027457
2215 015060 042074 043511 052111
2216 015066 046101 042452 052521
2217 015074 050111 042515 052116
2218 015102 041452 051117 047520
2219 015110 040522 044524 047117
2220 015116 030076 030060 030060
2221 015124 030060 030060 030060
2222 015132 030060 030060 030060
2223 015140 030060 030060 030060
2224 015146 030060 030060 030060
2225 015154 030060 030060 030060
2226 015162 030060 030060 030060
2227 015170 042074 043511 052111
2228 015176 046101 042452 052521
2229 015204 050111 042515 052116
2230 015212 041452 051117 047520
2231 015220 040522 044524 047117

```

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>=====

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>.....

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>////////////////////

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>00

.ASCII '<DIGITAL*EQUIPMENT*CORPORATION>11

D07

KL11 TESTS FOR VTOS DISPLAY TERMINAL
DZVTB.P11

MACY11 27(732) 20-SEP-76 10:11 PAGE 53

SEQ 0081

2344	016436	051505	020124	041101
2345	016444	051117	042524	040104
2346				
2347				

2404	017004	040043				
2405	017006	021445	044103	051101	CHRTST: .ASCII	'%#CHARACTER TESTS%#'
2406	017014	041501	042524	020122		
2407	017022	042524	052123	022523		
2408	017030	040043				
2409	017032	021445	047527	051522	WCPTST: .ASCII	'%#WORST CASE PATTERN TEST%#'
2410	017040	020124	040503	042523		
2411	017046	050040	052101	042524		
2412	017054	047122	052040	051505		
2413	017062	022524	040043			
2414	017066	021445	054513	042102	KMSG1: .ASCII	'%#KYBD TEST%#'
2415	017074	052040	051505	022524		
2416	017102	040043				
2417	017104	050045	042522	051523	KMSG2: .ASCII	'%PRESS A KEY WITHIN 10 SECONDS. '
2418	017112	040440	045440	054505		
2419	017120	053440	052111	044510		
2420	017126	020116	030061	051440		
2421	017134	041505	047117	051504		
2422	017142	040056				
2423	017144	021445	041505	047510	KMSG3: .ASCII	'%#ECHO TEST'
2424	017152	052040	051505	124	KMSG3A: .ASCII	'%CHARACTER KEYED WILL BE TYPED.'
2425	017157	045	044103	051101		
2426	017164	041501	042524	020122		
2427	017172	042513	042531	020104		
2428	017200	044527	046114	041040		
2429	017206	020105	054524	042520		
2430	017214	027104				
2431	017216	051045	041125	052517	.ASCII	'%#RUBOUT ENDS ROUTINE.%#'
2432	017224	020124	047105	051504		
2433	017232	051040	052517	044524		
2434	017240	042516	022456	040043		
2435	017246	021445	041517	040524	KMSG4: .ASCII	'%#OCTAL EQUIVALENT TEST#'
2436	017254	020114	050505	044525		
2437	017262	040526	042514	052116		
2438	017270	052040	051505	040124		
2439	017276	020045			KMSG5: .ASCII	'% '
2440	017300	020040	020040	040045	OCTEQV: .ASCII	'%#%#'
2441	017306	047045	020117	042513	KMSG6: .ASCII	'%#NO KEYBOARD REQUEST. '
2442	017314	041131	040517	042122		
2443	017322	051040	050505	042525		
2444	017330	052123	040056			
2445	017334	043045	046101	042523	KMSG7: .ASCII	'%#FALSE KYBD INTERRUPT#'
2446	017342	045440	041131	020104		
2447	017350	047111	042524	051122		
2448	017356	050125	040124			
2449	017362	021445	044504	050123	P7MG1: .ASCII	'%#DISPLAY EXERCISER%#'
2450	017370	040514	020131	054105		
2451	017376	051105	044503	042523		
2452	017404	022522	040043			
2453	017410	021445	054524	042520	P7MG2: .ASCII	'%#TYPE IN DATA :#'
2454	017416	044440	020116	040504		
2455	017424	040524	035040	100		
2456	017431	125	040040		BKSU: .ASCII	'U #'
2457	017434	020040	020040	040	DECVAL: .ASCII	' , '
2458	017441	000001			DEND: .END	

AT20C	006412	1073	1083#		
AT20D	006414	1082	1084#		
AT20E	006406	1081#			
AT21	006420	1069	1088#		
AT21A	006434	1094#			
AT21B	006444	1091	1097#	1103	
AT21C	006466	1101#	1107		
AT21E	006476	1093	1104#		
AT22	006504	1089	1113#		
AT22A	006530	1122#	1132		
AT22B	006564	1116	1118	1129#	
AT22C	006566	1128	1130#		
AT22E	006550	1127#			
AT23	006576	1114	1134#		
AT23A	006616	1137	1141#	1149	
AT23B	006640	1147#	1154	1157	
AT23C	006650	1142	1150#		
AT23D	006664	1151	1153#		
AT23E1	006636	1146#			
AT23E2	006670	1150	1155#		
AT24	006676	979	1159#		
AT24A	006714	1162	1164#		
AT24B	006736	1166	1169#		
AT24C	006756	1168	1171	1173#	1174
AT24E1	006732	1167#			
AT24E2	006754	1172#			
AT25	006762	1160	1176#		
AT25A	007000	1179	1181#	1187	
AT25B	007022	1184	1186#		
AT25E	007020	1185#			
AT26	007026	1177	1189#		
AT26A	007036	1192	1193#	1203	
AT26B	007060	1195	1198#		
AT26C	007100	1197	1200	1202#	
AT26E1	007054	1196#			
AT26E2	007076	1201#			
AT27	007104	1190	1206#		
AT27A	007114	1209	1210#	1216	
AT27B	007136	1213	1215#		
AT27E	007134	1214#			
AT3	005670	939	949#		
AT3A	005706	952	954#	956	
AT3B	005712	955#	958		
AT3E	005716	953	957#		
AT30	007142	1207	1218#		
AT30A	007152	1221	1222#	1226	
AT30B	007162	1223	1225#		
AT30E	007160	1224#			
AT31	007166	1219	1228#		
AT31A	007176	1231	1232#	1240	
AT31B	007220	1237	1239#		
AT31E	007216	1238#			
AT32	007224	1229	1242#		
AT32A	007234	1245	1246#	1256	
AT32B	007262	1253	1255#		
AT32E	007260	1254#			

F08

KL11 TESTS FOR VTOS DISPLAY TERMINAL MACY11 27(732) 20-SEP-76 10:11 PAGE 70
 DZV-B.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

SEG 0096

ADD	273 546 2000	296 549 2012	287 570	292 577	411 592	446 596	451 506	455 665	474 676	489 768	491 813	493 814	495 843	498 1507	500 1960
BOS	810														
BEQ	199 619 1200	221 627 1213	223 629 1432	240 631 1557	246 633 1751	296 635 1937	298 637 1970	346 651 1999	348 695 2026	355 704 2029	369 973 2039	391 596 2049	479 1057	488 1171	617 1194
BGT	649														
BHI	1995	2009													
BIC	315 1722 1857	333 1738 2036	351 1800	410 1822	445 1841	694 1844	703 1991	729 2005	738	767	846	971	1169	1198	1707
BICB	1857	2036													
BIS	244 1338 1923	556 1363	966 1390	983 1410	1079 1440	1099 1514	1125 1709	1144 1724	1164 1740	1181 1758	1193 1770	1210 1832	1283 1887	1300 1906	1320 1922
BISB	2024														
BIT	239 1194	328 1199	345 1212	352 1924	362	366	390	457	690	967	972	985	1165	1170	1183
BLE	371	639													
BLOS	317	406	418	1993	2007										
BMI	380	559	1010	1223	1253	1269	1415	1779	1851	2015					
BNE	192 684 1474 1943	329 691 1497 1890	335 727 1501 1925	340 736 1505 1961	353 774 1521 2002	363 784 1540	365 806 1549	367 833 1561	450 968 1569	454 1014 1727	458 1062 1743	563 1166 1787	610 1195 1802	668 1452 1824	681 1458 1840
BPL	242 1928	258 1933	344 1967	389	647	789	1027	1041	1237	1795	1799	1815	1854	1908	1912
BR	233 621 958 1128 1287 1508	261 641 970 1132 1289 1524	288 643 976 1149 1304 1777	309 654 989 1154 1307 1852	319 661 999 1157 1323 1869	331 672 1012 1168 1327 1885	338 792 1016 1174 1343 1915	342 812 1030 1187 1348 1939	373 923 1044 1197 1351 1971	375 925 1060 1203 1367	387 934 1064 1216 1371	408 936 1082 1226 1398	420 945 1085 1240 1420	504 947 1103 1256 1436	565 956 1107 1272 1463
CLR	304 1130 1364	306 1143 1369	311 1147 1389	313 1235 1391	320 1265 1396	399 1281 1411	557 1282 1494	678 1299 1536	796 1302 1771	808 1317 1813	1077 1325 1982	1078 1336	1097 1337	1101 1341	1122 1362
CLRB	1251														
CMP	191 628	198 648	222 650	245 667	295 1823	297 1969	316 1992	339 1994	347 2006	364 2008	368 2038	370	453	478	487
CMPB	220 1842	334	405	417	609	616	618	626	628	630	632	634	636	1056	1901
COM	584	725	726	734	735	1953	1956	1959	1990	2004					
DEC	354 1486 2048	562 1500	640 1504	652 1520	680 1539	683 1548	773 1560	783 1568	805 1726	832 1742	1013 1796	1061 1939	1451 1968	1457 2001	1473 2025
EMT	95 111	96 112	97 113	98	99	100	101	102	103	104	105	106	108	109	110
HALT	32 431	34 1059	36 2031	38	40	48	53	195	217	235	264	270	374	407	419
INC	308	728	737	811	1250	1538	1891	2037	2052	2053					
JMP	125 1761	214 1983	276 2027	302 2034	323 2040	326 2044	394 2047	401 2050	413 2054	441	448	484	510	914	1443
JSR	201 359 746 873	205 360 753 877	210 372 754 883	213 480 794 887	224 485 795 891	228 506 804 898	232 612 831 902	248 614 835 906	252 620 837 997	257 658 844 1007	260 660 847 1024	293 693 854 1038	327 702 858 1052	330 711 862 1076	336 745 969 1096

H08

KL11 TESTS FOR VTOS DISPLAY TERMINAL MACY11 27(732) 20-SEP-76 10:11 PAGE 72
DZVTB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

SEQ 0098

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DZVTB/SOL/CRF.PAGNUM=DZVTB
RUN-TIME: 2 16 4 SECONDS
RUN-TIME RATIO: 109/30=3.6
CORE USED: 11K (21 PAGES)

