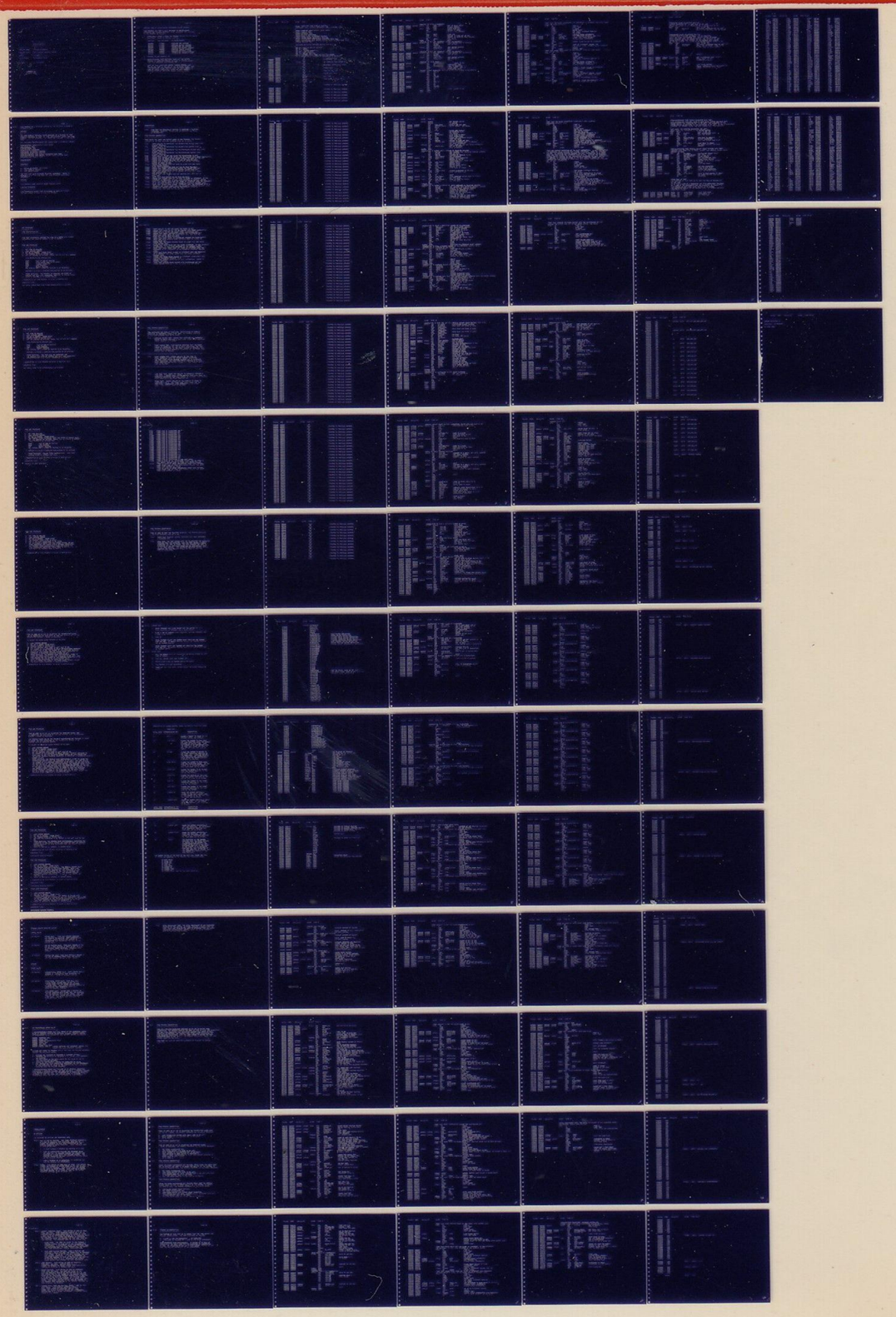
VT06

DISPLAY TERMINAL TESTS
MD-11-D6D-B

EP-D6D-DL

FICHE 1 OF 1

FEB 1978
digital
MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-D6DB-D

PRODUCT NAME: VT86 DISPLAY TERMINAL TESTS

DATE: JULY,1971

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: J. FRIEDRICH

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

1. ABSTRACT

THE VT06 DISPLAY TERMINAL TESTS CONSISTS OF A PACKAGE OF TEST PROGRAMS DESIGNED TO TEST THE VT06 INPUT-OUTPUT LOGIC, THE VT06 DISPLAY, AND THE KEYBOARD. ALL TESTS ARE INCLUDED IN ONE OBJECT TAPE.

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

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11,20 SYSTEM, (4 K CORE),
B. VT06 DISPLAY TERMINAL
C. HIGH SPEED READER

THE VT06 MUST HAVE STANDARD TELETYPE ADDRESSES. REFER TO SECTION 7,3 IF THE VT06 DOES NOT HAVE STANDARD PERIPHERAL ADDRESSES.

2.2 STORAGE

THIS PROGRAM USES LOCATION 00200 THROUGH 11672.

3. LOADING PROCEDURE

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

4, USE PROCEDURE

4,1 VT8Ø IDENTIFICATION

THIS TEST DIAGNOSTIC ASSUMES THE VT86 IS A KSR35 WITH CURSOR CHARACTERS AND LOCATION 000224 IS SET TO 000001,

4,2 PRGØ USE PROCEDURE

A, SET VT86 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,

4,3 PRG1 USE PROCEDURE

A, SET VT86 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
SR0 THROUGH SR8	NUMBER OF ROUTINE TO BE SELECTED.

SECTION 7,2 GIVES A COMPLETE EXPLANATION OF SR OPTIONS.

F, PRESS CONTINUE, THE VT86 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 8,3

EXECUTION TIME:

ONE NORMAL PASS TAKES APPROXIMATELY 12 MINUTES.

4.2 PRG2 USE PROCEDURE

A. SET VT86 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 SR8 = 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 8.5

EXECUTION TIME:

PROGRAM IS USER DEPENDENT.

4.3 PRG3 USE PROCEDURE

A. SET VT86 TO ON-LINE
B. LOAD ADDRESS 000200
C. SET SR TO 000003, PRESS START
D. THE PROGRAM TYPES "TYPE IN DATA"
E. KEY IN ANY FIVE CHARACTERS TO BE TYPED.
F. KEY IN EITHER A RUBOUT FOR FULL SPEED TYPING, OR ANY OTHER CHARACTER FOR RANDOM STALLS BETWEEN CHARACTERS.
G. THE PROGRAM TYPES CONTINUOUSLY LINES CONTAINING THE FIVE CHARACTERS SPECIFIED, UNTIL SR15 IS SET TO A 1. AT THAT POINT THE PROGRAM GOES TO STEP E.

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8.8

4,4 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 8,18

4.5 PRG9 USE PROCEDURE

PRG9 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 HE 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 8.11

4,6 PRG6 USE PROCEDURE

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

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8,12

EXECUTION TIME:

CONTINUOUS RUNNING PROGRAM,

4,7 PRG7 USE PROCEDURE

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

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8,13

EXECUTION TIME:

CONTINUOUS RUNNING PROGRAM,

4,10 PRG10 USER PROCEDURE

A, LOAD ADDRESS 000200,
B, SET SR TO 000007, PRESS START
C, PROGRAM RUNS CONTINUOUSLY, THE SCREEN IS FILLED WITH ALTERNATE LINES OF A CHARACTER AND ITS COMPLEMENT AND A LINE OF THE COMPLEMENT OF THE CHATACTER FOLLOWED BY THE CHARACTER, THIS TEST VERIFIES THE ROLL-UP CAPABILITY OF THE VT06,
D, REFER TO SECTION 6, ERRORS, IF ERRORS OCCUR,

A DESCRIPTION OF THIS PROGRAM IS GIVEN IN SECTION 8,14

EXECUTION TIME:

CONTINUOUS RUNNING PROGRAM,

5. PROGRAM AND/OR OPERATOR ACTION

5.1 NORMAL HALTS

LOC 001374 — 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 001504 — 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 002012 — PROGRAM END HALT, THIS HALT NORMALLY OCCURS AT THE END OF PROGRAMS PRG0, PRG1 AND UNLESS THE LOOP PROGRAM OPTION IS SET, (SR12)

6. ERRORS

6.1 ERROR HALTS

LOC 001406 — UNCONDITIONAL ERROR HALT. DATA LIGHTS CONTAIN ADDRESS OF INSTRUCTION THAT GENERATED THE ERROR CALL. REFER TO PROGRAM LISTING.

LOC 001466 — 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 001426 — 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.

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 000196.

000002 RESERVED AREA
000006 ERROR TRAP
000012 RESERVED INSTRUCTION TRAP
000016 DEBUG TRAP
000022 IOT TRAP
000026 POWER FAIL TRAP
000048 THROUGH 000176 SYSTEM SOFTWARE AND INTERRUPT VECTOR AREA, EXCEPT FOR KL11 INTERRUPT VECTORS,

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

A, EXAMINE THE CONTENTS OF REGISTER 6 (ADDRESS 177706),
B, TRANSFER THE CONTENTS OF REGISTER 6 TO THE SR, LOAD ADDRRESS, AND EXAMINE,
C, THE DATA SHOWN IN THE DATA LIGHTS IS THE VALUE OF THE PC WHEN THE FAILURE OCCURRED,
D, LOCATE IN THE PROGRAM LISTING THE DISPLAYED PC VALUE,
E, THE INSTRUCTION THAT IMMEDIATELY PRECEDES 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 VT06, ANY FURTHER STEPS TO DIAGNOSE A NON-RECOVERABLE ERROR ARE NOT WITHIN THE SCOPE OF THIS PROGRAM,

7. MISCELLANEOUS

7.1 SR OPTIONS

THE STANDARD SR OPTIONS ARE DESCRIBED HERE.

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

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

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

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

SR14 - SCOPE. THIS OPTION IS USED ONLY BY PRGB. 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)

SR13 - 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 SR13 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.

SR12 - LOOP PROGRAM. THIS OPTION IS USED BY PROGRAMS PRG0,PRG1, AND PRG4. SETTING SR12 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 8 TO OBTAIN THE ROUTINE NUMBER. THE ROUTINE NUMBER SELECTED MUST BE A VALID NUMBER, OR AN ERROR HALT WILL OCCUR. THE SELECT ROUTINE OPTION WILL BE HONORED BY THE PROGRAM UPON COMPLETION OF THE CURRENT ROUTINE, OR UPON STARTING THE PROGRAM.

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

7,3 TESTING VT06 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
001206	177560	RECEIVER CSR ADDRESS
001210	177562	RECEIVER BUFFER ADDRESS
001212	177564	TRANSMITTER CSR ADDRESS
001214	177566	TRANSMITTER BUFFER ADDRESS
001216	000060	RECEIVER VECTOR ADDRESS
001220	000200	RECEIVER PRIORITY LEVEL
001222	000064	TRANSMITTER VECTOR ADDRESS
001224	000200	TRANSMITTER PRIORITY LEVEL

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.

8. DESCRIPTION

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

8.1 PRG0 PROGRAM DESCRIPTION

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

RTN0 TESTS ABILITY TO REFERENCE THE READER/KYBD STATUS WORD (TKS) WITHOUT TRAPPING.
RTN1 TESTS ABILITY TO REFERENCE THE READER/KYBD BUFFER (TKB) WITHOUT TRAPPING.
RTN2 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH STATUS WORD (TPS) WITHOUT TRAPPING.
RTN3 TESTS ABILITY TO REFERENCE THE PRINTER/PUNCH BUFFER (TPB) WITHOUT TRAPPING.
RTN4 TESTS ABILITY TO SET AND CLEAR THE READER/KYBD ID BIT.
RTN5 CHECKS THAT READER /KYBD ID BIT CAN BE CLEARED WITH RESET INSTRUCTION.
RTN6 CHECKS THAT 30 MSECS AFTER READER ENABLE THE BUSY BIT HAS SET.
RTN7 CHECKS THAT READER/KYBD BUSY BIT CAN BE READ RELIABLY.
RTN10 CHECKS THAT 8 MSECS AFTER THE BUSY BIT HAS SET THE READ BUFFER IS RESET TO 200.
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.
RTN15 TESTS THAT READER DONE BIT IS CLEARED BY START BIT. (WHEN BUSY BECOMES SET.)
RTN16 CHECKS THAT WHEN READER DONE BIT BECOMES SET, BUSY BIT IS STILL SET.
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 S_T.

(8,1 CONT'D)

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

8,2 PRG1 PROGRAM DESCRIPTION

PRG1 EXERCISES THE DISPLAY FUNCTIONS, VERIFICATION OF DISPLAY OPEMATION IS PERFORMED VISUALLY BY USER, THE PROGRAM CONTAINS 29 HOUTINES NUMBERED FROM 00 TO 34(8),

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

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

```
----I----I----I----I----I----I----I----I----I----I----I----I----I----I-I
```

RTN2 RIGHT CURSOR TEST, CHECKS ABILITY OF THE VT06 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,

RTN4 SPACE TEST, CHECKS ABILITY OF THE DISPLAY TO SPACE TO POSITIONS 1 TO 7, THE FIRST LINE TYPED MARKS THE TAB POSITIONS, SUBSEQUENT LINES INCREMENT BY ONE SPACE EACH, THE TYPEOUT LOOKS AS FOLLOWS:

```
                /       /       /       /       /       /       /       /

         //////////

          / / / / / / / / / /

           /  /  /  /  /  /  /  /  /  /

            /   /   /   /   /   /   /   /   /

             /    /    /    /    /    /    /    /    /

              /     /     /     /     /     /     /     /     /
```

(8',2 CONT'D)

```
        RTN5    TYPES LINE OF CHARACTERS ABC
        RTN6    TYPES LINE OF CHARAC"ERS 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 Y@@
        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 1;<
        RTN27   TYPES LINE OF CHARACTERS =>?
        RTN30   TYPES LINE OF CHARACTERS @[\
        RTN31   TYPES LINE OF CHARACTERS ]↑ AND LEFT ARROW
        RTN32   TYPES 2  INES OF ALL CHARACTERS, FIRST LINE IS TYPED AT
                FULL SPEED, SECOND LINE IS TYPED WITH RANDOM STALLS.
        RTN33   TYPES 12 LINES OF ASR33 (@@1224@18) 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 (@@1224@11) WORST CASE PATTERN,
                EVERY OTHER LINE IS TYPED WITH RANDOM STALLS, THE ASR35
                WORST CASE PATTERN IS '[?C?[
```

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 VT8A CONTROL. THE FOLLOWING SECTIONS (8.3.1, 8.3.2) DESCRIBE HOW THIS ROUTINE SHOULD BE USED TO TEST THE SPECIAL CHARACTERS,

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 23 TIMES AND SCREEN SHOULD BE CLEAR.

D. FILL SCREEN WITH ANY CHARACTER.

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

DESCRIPTION OF CURSOR CONTROL CODES AND SPECIAL FUNCTION CODES

TABLE 8-3

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

OCTAL CODE | CORRESPONDING KEY | DESCRIPTION

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

THE CURSOR SWITCH AT THE REAR OF THE UNIT WILL CAUSE THE VT86 TO IGNORE (NO RESPONSE ON THE SCREEN) THE FOLLOWING CODES

10 BACK SPACE
37 ERASE EOP
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.

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.

8.8 PRGJ PROGRAM DESCRIPTION

PRGJ IS A PRINTER EXERCISER DESIGNED AS AN AID IN MAKING VT86 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 B,
E. ISSUE 9 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,

8.18 PROGRAM 18 DESCRIPTION

THE PURPOSE OF THIS TEST IS TO VERIFY THAT THE VT86 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 18 CAN TEST THE ROLL-UP CAPABILITY OF ALL CHARACTERS.

```
                                ;KL11  TESTS FOR VT06 DISPLAY TERMINAL
                                ;COPYRIGHT 1971, 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

                                ;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,
        000000                          .=0
000000  000002                          .+2                     ;UNASSIGNED TRAP
000002  000000                          HALT
000004  000006              MACHERR:    .+2                     ;SP OVERFLOW, BUS ERROR TRAP
000006  000000                          HALT
000010  000012                          .+2                     ;RESERVED INSTRUCTION TRAP
000012  000000                          HALT
000014  000016                          .+2                     ;TRACE TRAP
000016  000000                          HALT
000020  000022                          .+2                     ;TRAP TO CALL IOX
000022  000000                          HALT
000024  000026                          .+2                     ;POWER FAIL TRAP
000026  000000                          HALT
000030  002146                          EMTINT                  ;EMT TRAP
000032  000340                          PRTY7
000034  002212                          TRPINT                  ;TRAP TRAP, SIMILAR TO EMT
000036  000340                          PRTY7
        000200                          .REPT   200
                                        .+2
                                        HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
                                        .ENDR
000040  000042                          .+2
000042  000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000044  000046                          .+2
000046  000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000050  000052                          .+2
000052  000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000054  000056                          .+2
000056  000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000060  000062                          .+2
000062  000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000064  000066                          .+2
```

```
000066  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000070  000072          .+2
000072  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000074  000076          .+2
000076  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000100  000102          .+2
000102  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000104  000106          .+2
000106  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000110  000112          .+2
000112  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000114  000116          .+2
000116  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000120  000122          .+2
000122  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000124  000126          .+2
000126  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000130  000132          .+2
000132  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000134  000136          .+2
000136  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000140  000142          .+2
000142  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000144  000146          .+2
000146  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000150  000152          .+2
000152  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000154  000156          .+2
000156  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000160  000162          .+2
000162  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000164  000166          .+2
000166  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000170  000172          .+2
000172  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000174  000176          .+2
000176  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000200  000202          .+2
000202  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000204  000206          .+2
000206  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000210  000212          .+2
000212  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000214  000216          .+2
000216  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000220  000222          .+2
000222  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000224  000226          .+2
000226  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000230  000232          .+2
000232  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000234  000236          .+2
000236  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000240  000242          .+2
000242  000000          HALT            ;TRAPPED TO PREVIOUS ADDRESS.
```

```
000244  000246                  ,+2
000246  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000250  000252                  ,+2
000252  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000254  000256                  ,+2
000256  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000260  000262                  ,+2
000262  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000264  000266                  ,+2
000266  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000270  000272                  ,+2
000272  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000274  000276                  ,+2
000276  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000300  000302                  ,+2
000302  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000304  000306                  ,+2
000306  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000310  000312                  ,+2
000312  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000314  000316                  ,+2
000316  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000320  000322                  ,+2
000322  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000324  000326                  ,+2
000326  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000330  000332                  ,+2
000332  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000334  000336                  ,+2
000336  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000340  000342                  ,+2
000342  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000344  000346                  ,+2
000346  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000350  000352                  ,+2
000352  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000354  000356                  ,+2
000356  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000360  000362                  ,+2
000362  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000364  000366                  ,+2
000366  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000370  000372                  ,+2
000372  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000374  000376                  ,+2
000376  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000400  000402                  ,+2
000402  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000404  000406                  ,+2
000406  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000410  000412                  ,+2
000412  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000414  000416                  ,+2
000416  000000                  HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000420  000422                  ,+2
```

```
000422  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000424  000426          .+2
000426  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000430  000432          .+2
000432  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000434  000436          .+2
000436  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000440  000442          .+2
000442  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000444  000446          .+2
000446  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000450  000452          .+2
000452  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000454  000456          .+2
000456  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000460  000462          .+2
000462  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000464  000466          .+2
000466  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000470  000472          .+2
000472  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000474  000476          .+2
000476  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000500  000502          .+2
000502  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000504  000506          .+2
000506  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000510  000512          .+2
000512  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000514  000516          .+2
000516  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000520  000522          .+2
000522  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000524  000526          .+2
000526  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000530  000532          .+2
000532  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000534  000536          .+2
000536  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000540  000542          .+2
000542  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000544  000546          .+2
000546  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000550  000552          .+2
000552  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000554  000556          .+2
000556  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000560  000562          .+2
000562  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000564  000566          .+2
000566  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000570  000572          .+2
000572  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000574  000576          .+2
000576  000000          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
```

```
000000  000002                  .+2
000002  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000004  000006                  .+2
000006  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000010  000012                  .+2
000012  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000014  000016                  .+2
000016  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000020  000022                  .+2
000022  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000024  000026                  .+2
000026  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000030  000032                  .+2
000032  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000034  000036                  .+2
000036  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000040  000042                  .+2
000042  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000044  000046                  .+2
000046  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000050  000052                  .+2
000052  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000054  000056                  .+2
000056  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000060  000062                  .+2
000062  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000064  000066                  .+2
000066  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000070  000072                  .+2
000072  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000074  000076                  .+2
000076  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000100  000102                  .+2
000102  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000104  000106                  .+2
000106  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000110  000112                  .+2
000112  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000114  000116                  .+2
000116  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000120  000122                  .+2
000122  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000124  000126                  .+2
000126  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000130  000132                  .+2
000132  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000134  000136                  .+2
000136  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000140  000142                  .+2
000142  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000144  000146                  .+2
000146  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000150  000152                  .+2
000152  000000                  HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
000154  000156                  .+2
```

```
000756  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000760  000762          .+2
000762  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000764  000766          .+2
000766  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000770  000772          .+2
000772  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
000774  000776          .+2
000776  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001000  001002          .+2
001002  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001004  001006          .+2
001006  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001010  001012          .+2
001012  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001014  001016          .+2
001016  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001020  001022          .+2
001022  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001024  001026          .+2
001026  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001030  001032          .+2
001032  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
001034  001036          .+2
001036  000000          HALT                ;TRAPPED TO PREVIOUS ADDRESS.
```

```
                        ;EQUATE STATEMENTS
177570                  SR=177570
177776                  CC=177776
177776                  PSW=177776
000240                  NOP=240
000000                  OPEN=0
100000                  HLTSW=BIT15             ;HALT SWITCH DEFINITION
040000                  SCOPSW=BIT14            ;SCOPE SWITCH DEFINITION
020000                  NPRTSW=BIT13            ;INHIBIT PRINT SWITCH DEFINITION
010000                  NTRCSW=BIT12            ;INHIBIT TRACE SWITCH DEFINITION
004000                  NITRSW=BIT11            ;INHIBIT ITERATION SWITCH DEFINITION
002000                  LPPGSW=BIT10            ;LOOP PROGRAM SWITCH DEFINITION
001000                  SRTSW=BIT9              ;SELECT ROUTINE SWITCH DEFINITION
000400                  BYPMAN=BIT8             ;BYPASS MANUAL INTERVENTION DEFINITION.
100000                  MANUAL=BIT15
100000                  BIT15=100000
040000                  BIT14=40000
020000                  BIT13=20000
010000                  BIT12=10000
004000                  BIT11=4000
002000                  BIT10=2000
001000                  BIT9=1000
000400                  BIT8=400
000200                  BIT7=200
000100                  BIT6=100
000040                  BIT5=40
000020                  BIT4=20
000010                  BIT3=10
000004                  BIT2=4
000002                  BIT1=2
000000                  BIT0=0
005726                  POPSP=5726              ;POP THE STACK, SAME AS TST (6)+
022626                  POPSP2=022626           ;POP STACK TWICE, SAME AS CMP (6)+,(6)+
000340                  PRTY7=340               ;PRIORITY LEVEL DEFINITIONS
000300                  PRTY6=300
000240                  PRTY5=240
000200                  PRTY4=200
000140                  PRTY3=140
000100                  PRTY2=100
000040                  PRTY1=40
000000                  PRTY0=0
104000                  TYPE=EMT+0
104001                  TYPES=EMT+1
104002                  STALL=EMT+2
104003                  ERROR=EMT+3
104004                  DATCHK=EMT+4
104005                  CHALT=EMT+5
104006                  STRDRV=EMT+6
104007                  STPCHV=EMT+7
104010                  EHALT=EMT+10
104011                  SRESET=EMT+11
104012                  CHAIN=EMT+12
104013                  CK33=EMT+13
```

```
                104014                          CK35=EMT+14
                104016                          TYPLN3=EMT+16
                104017                          DATHLT=EMT+17
                104020                          SAVREG=EMT+20
                104021                          RSTREG=EMT+21
                104022                          CHKASR=EMT+22
                104400                          DELAY=TRAP+0
                000007                          BELL=007
                011671                          BLOCKA=.END
                011673                          BLOCK1=BLOCKA+2
                012003                          BLOCKB=BLOCKA+112
                012014                          BLKBB=BLOCKA+123
                012005                          BLOCK2=BLOCKA+114
                012016                          BLK2=BLOCKA+125
                012115                          BLOCKC=BLOCKA+224
                012126                          BLKCC=BLOCKA+235

                000200                          .=200
000200  000167  001204                          JMP     START           ;GO TO START OF PROGRAM.
                001204                  .=.+1000                        ;GET CODE OUT OF VECTOR AREA
001204  000000                  SPBOT:  0                               ;BOTTOM OF STACK
001206  177560                  TKS:    177560                          ;LSR CSR
001210  177562                  TKB:    177562                          ;LSR BUFFER
001212  177564                  TPS:    177564                          ;LSP CSR
001214  177566                  TPB:    177566                          ;LSP BUFFER
001216  000060                  TKVTR:  60                              ;LSR INTERRUPT VECTOR
001220  000200                  TKLVL:  PRTY4                           ;LSR PRIORITY LEVEL
001222  000064                  TPVTR:  64                              ;LSP INTERRUPT VECTOR
001224  000200                  TPLVL:  PRTY4                           ;LSP PRIORITY LEVEL
001226  000001                  TTYTYP: 01              ;TTY = KSR35
001230  000000                  PRGNUM: OPEN                            ;CONTAINS CURRENT PROGRAM#
001232  000000                  KSTART: OPEN                            ;CURRENT PROGRAM START ADDRESS.
001234  000000                  CURTST: OPEN                            ;CONTAINS ADDR OF CURRENT TEST.
001236  000000                  RTNNO:  OPEN                            ;CONTAINS CURRENT TEST #.
001240  000000                  NXTST:  OPEN                            ;CONTAINS ADDR OF NEXT TEST.
001242  000000                  ICTR:   OPEN                            ;CONTAINS CURRENT ITERATION COUNT
001244  000000                  SCOPTR: OPEN                            ;CONTAINS CURRENT SCOPE POINTER.
001246  000000                  PRGID:  OPEN                            ;CONTAINS PROGRAM INDICATORS
001250  004526                  PRGTAB: PRG0                            ;PRG0 START ADDRESS
001252  006210                          PRG1                            ;PRG1 START ADDRESS
001254  007604                          PRG2                            ;PRG2 START ADDRESS
001256  010104                          PRG3                            ;PRG3 START ADDRESS
001260  010264                          PRG4                            ;PRG4 START ADDRESS
001262  010274                          PRG5                            ;PRG5 START ADDRESS
001264  010364                          PRG6                            ;PRG6 START ADDRESS
001266  010436                          PRG7                            ;PRG7 START ADDRESS
001270  010540                          PRG10                           ;PRG10 START ADDRESS
```

```
001272  002744          EMTTAB: TYP             ;POINTER TO TYPEOUT ROUTINE
001274  003066                  TYPS            ;POINTER TO CHAINED MESSAGES ROUTINE
001276  003164                  STAL            ;POINTER TO RANDOM STALL ROUTINE
001300  001432                  ERR             ;POINTER TO ERROR ROUTINE
001302  001412                  DTCHK
001304  001366                  CHLT            ;COMMON HALT
001306  002474                  STLSRV
001310  002524                  STLSPV
001312  001400                  EHLT            ;POINTER TO ERROR HALT ROUTINE;
001314  002554                  SRSETT
001316  001672                  CHAINN
001320  002256                  CHK33
001322  002272                  CHK35
001324  002270                  CHK330
001326  004244                  TYPL3
001330  001422                  DTHLT
001332  002334                  SAVRG
001334  002374                  RSTRG
001336  002310                  CKASR
001340  003120          TRPTAB: DLY
001342  000000          RCNT:   OPEN            ;CHARACTER COUNT
001344  000000          CRBUF:  OPEN            ;HOLDS ONE CHARACTER FROM READER.
001346  000000          CHR1:   OPEN
001350  000000          CHR2:   OPEN
001352  000000          CHR3:   OPEN
001354  000000          ERCTR:  OPEN
001356  000000          CTRA:   OPEN
001360  000000          CTRB:   OPEN
001362  000000          CTRC:   OPEN
001364  000000          CTRD:   OPEN
```

```
                                ;COMMON HALT ROUTINE
001366  011600                  CHLT:   MOV     @X6,X0          ;DEVELOP ADDRESS OF CALLER,
001370  162700  000002                  SUB     #2,X0
001374  000000                          HALT                    ;HALT, ADDRESS OF CALL INSTRUCTION
001376  000002                          RTI                     ;IN DATA LIGHTS,
                                ;UNCONDITIONAL ERROR HALT ROUTINE,
001400  011600                  EHLT:   MOV     @X6,X0          ;DEVELOP ADDRESS OF CALLER,
001402  162700  000002                  SUB     #2,X0
001406  000000                          HALT                    ;HALT, ADDR OF ERROR CALL
001410  000002                          RTI                     ;IN DATA LIGHTS,
                                ;DATA CHECK ROUTINE,
001412  126767  177726  177725  DTCHK:  CMPB    CRBUF,CRBUF+1   ;COMPARE EXPECTED AND RECEIVED
001420  001403                          BEQ     DTCHKA          ;CHARS, BRANCH IF SAME,
001422  016700  177716          DTHLT:  MOV     CRBUF,X0        ;MOVE S/B AND WAS CHARS TO R0,
001426  000000                          HALT                    ;DATA ERROR HALT, GOOD CHAR IN
                                                                ;DATA LIGHTS LEFT, BAD CHAR IN DATA
001430  000002                  DTCHKA: RTI                     ;LIGHTS RIGHT, EXIT,
                                ;CONDITIONAL ERROR HALT,
001432  032767  040000  176130  ERR:    BIT     #SCOPSW,SR      ;CHECK SCOPE SWITCH,
001440  001404                          BEQ     ERRA            ;BRANCH IF NO SCOPE DESIRED,
001442  005767  177600                  TST     PRGID           ;SCOPING WANTED, FIRST ERROR?
001446  100001                          BPL     ERRA            ;NO SCOPE IF FIRST ERROR,
001450  000002                          RTI                     ;SCOPE EXIT,
001452  052767  100000  177566  ERRA:   BIS     #BIT15,PRGID    ;SET ERROR INDICATOR,
001460  011600                          MOV     @X6,X0          ;DEVELOP CALLER'S ADDRESS,
001462  162700  000002                  SUB     #2,X0
001466  000000                          HALT                    ;ERROR HALT,
001470  000002                          RTI                     ;EXIT,
                                ;ROUTINE END HALT SUBROUTINE,
001472  005767  176872          SHALT:  TST     SR              ;CHECK HALT SWITCH,
001476  100003                          BPL     SHLTA           ;BRANCH IF NO HALT DESIRED,
001500  116700  177532                  MOVB    RTNNO,X0        ;CURRENT TEST # TO R0,
001504  000000                          HALT                    ;ROUTINE END HALT,
001506  000207                  SHLTA:  RTS     X7              ;EXIT,
```

```
001910  012706  001204          START:  MOV     #SPBOT,%6       ;SET BOTTOM OF SP STACK.
001914  005067  176256                  CLR     PSW
001920  012767  000006  176256          MOV     #6,MACHER
001926  005067  177504                  CLR     RTNNO
001932  016700  176032                  MOV     SR,%0           ;(SR) TO R0
001936  042700  177760                  BIC     #177760,%0      ;LIMIT (SR) TO BITS 3-0
001942  020027  000014                  CMP     %0,#14          ;COMPARE (SR) TO PROGRAM LIMIT
001946  101402                          BLOS    CRTA            ;VALID PROGRAM NUMBER?
001950  104010                  INCPRG: EMALT                   ;NO, ;INCORRECT PRG NUMBER
001952  000756                          BR      START           ;START OVER.
001954  005067  177466          CRTA:   CLR     PRGID
001960  010067  177444                  MOV     %0,PRGNUM       ;SAVE PROGRAM NUMBER AT PRGNUM
001964  006100                          ROL     %0              ;R0X2
001966  000170  001250                  JMP     @PRGTAB(0)      ;GO TO SELECTED PROGRAM.
001972  104005                  SRSET:  CMALT                   ;SET SR OPTIONS DESIRED
001974  016767  177432  177436  GETRDY: MOV     KSTART,NXTST    ;ADDR OF 1ST ROUTINE TO NXTST
002002  000167  000314                  JMP     CLEAN           ;GO CLEAN UP.
002006  004767  000204          GTRDYA: JSR     %7,FORWD        ;ROLL FORWARD TO "NEXT" ROUTINE.
002012  032767  001000  175750  GTRDYB: BIT     #SRTSW,SR       ;CHECK FOR SELECT ROUTINE SWITCH
002020  001003                          BNE     GTRDYC          ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
002022  004767  000240                  JSR     %7,GOTST        ;GO RUN CURRENT ROUTINE.
002026  000455                          BR      CHNB            ;NO GO, MANUAL RTN BYPASSED.
002030  016700  175734          GTRDYC: MOV     SR,%0           ;(SR) TO R0
002034  042700  177000                  BIC     #177000,%0      ;MASK UNDESIRED BITS
002040  126700  177372                  CMPB    RTNNO,%0        ;COMPARE RTNNO TO (R0)
002044  001004                          BNE     GTRDYD          ;BRANCH IF ROUTINE NOT FOUND YET.
002046  004767  000222                  JSR     %7,GOTST        ;GO RUN ROUTINE.
002052  104010                          EMALT                   ;NO GO, MANUAL RTN SELECTED BYPASSED.
002054  000747                          BR      GETRDY
002056  022767  177777  177354  GTRDYD: CMP     #-1,NXTST       ;NO, CHECK FOR LAST ROUTINE.
002064  001350                          BNE     GTRDYA          ;LAST ROUTINE?
002066  104010                  INCRTN: EMALT                   ;YES, ;INCORRECT ROUTINE SELECTED.
002070  000741                          BR      GETRDY          ;START OVER.
002072  005767  177350          CHAINN: TST     PRGID           ;TEST ERROR BIT IN PRGID.
002076  100013                          BPL     CHNA            ;BRANCH IF ERROR BIT NOT SET.
002100  032767  040000  175662          BIT     #SCOPSW,SR      ;ERROR BIT SET, CHECK FOR SCOPE OPTION.
002106  001407                          BEQ     CHNA            ;SCOPE SWITCH SET IN SR?
002110  022767  177777  177326          CMP     #-1,SCOPTR      ;YES, CHECK SCOPE ENTRY POINTER
002116  001403                          BEQ     CHNA            ;BRANCH IF SCOPE ENTRY IS -1.
002120  017716  177320                  MOV     @SCOPTR,@%6     ;SET UP TO GO SCOPING
002124  000002                          RTI                     ;GO TO SCOPE ENTRY.
002126  042767  100000  177312  CHNA:   BIC     #BIT15,PRGID    ;CLEAR ERROR BIT IN PRGID.
002134  032767  004000  175626          BIT     #NITRSW,SR      ;TEST INHIBIT ITERATION SWITCH
002142  001004                          BNE     CHNAA           ;INHIBIT ITERATION?
002144  005367  177272                  DEC     ICTR            ;NO
002150  001401                          BEQ     CHNAA           ;COUNT 0?
002152  000002                          RTI                     ;NO, RETURN TO TEST ROUTINE
002154  022626                  CHNAA:  POPSP2                  ;POP STACK TWICE
002156  004767  177510                  JSR     %7,SHALT        ;GO HALT IF HALT SWITCH IS SET
```

```
002162  032767  001000  175600  CHNB1:  BIT     #SRTSW,SR       ;CHECK SELECT ROUTINE SWITCH
002166  001301                          BNE     GETRDY          ;SELECT ROUTINE SWITCH SET?
002170  022767  177777  177240          CMP     #-1,NXTST       ;NO,
002176  001306                          BNE     GTRDYA-4        ;LAST TEST?
002200  032767  002000  175560          BIT     #LPRGSW,SR      ;YES, TEST LOOP PROGRAM SWITCH,
002206  001271                          BNE     GETRDY          ;LOOP PROGRAM?
002210  000000                  PRGEND1:MALT                    ;NO, PROGRAM END.
002212  000762                          BR      CHNB
002214  016705  177216          FORWD1: MOV     NXTST,R5        ;ADDR OF NEXT ROUTINE TO R5,
002220  012567  177216                  MOV     (5)+,RTNNO      ;GET NEXT ROUTINE NUMBER,
002224  012567  177206                  MOV     (5)+,NXTST      ;GET ADDR OF NEXT "NEXT" ROUTINE,
002230  105767  177210                  TSTB    PRGID           ;CHECK IF PROGRAM SCOPE AND 1 COUNT
002234  100407                          BMI     FORWDB          ;PARAMETERS, BRANCH IF NOT,
002236  012567  177176                  MOV     (5)+,ICTR       ;GET ITERATION COUNT,
002242  012567  177174                  MOV     (5)+,SCOPTR     ;GET SCOPE LOOP ENTRY POINTER,
002246  010567  177160          FORWDA1:MOV     R5,CURTST       ;ADDR OF NOW CURRENT TEST TO CURTST,
002252  000207                          RTS     PC              ;EXIT FORWD SUBROUTINE,
002254  012767  177777  177160  FORWDB1:MOV     #-1,SCOPTR      ;FORCE "NO SCOPE"
002262  012767  000001  177150          MOV     #1,ICTR         ;FORCE 1 COUNT OF 1
002270  000766                          BR      FORWDA
002272  005767  177136          GOTST1: TST     RTNNO           ;CHECK FOR MANUAL RTN,
002276  100005                          BPL     GOTSTA          ;BRANCH IF NOT MANUAL RTN,
002300  032767  000400  175460          BIT     #BYPMAN,SR      ;MANUAL RTN, BYPASS IT?
002306  001401                          BEQ     GOTSTA          ;NO, RUN IT,
002310  000207                          RTS     PC              ;BYPASS MANUAL ROUTINE,
002312  005726                  GOTSTA1:POPSP
002314  000177  177112                  JMP     @CURTST         ;GO RUN TEST,
002320  012767  000006  175654  CLEAN1: MOV     #6,MACHER       ;RESET MACHER TRAP,
002326  005067  175642                  CLR     PSW
002332  012706  001204                  MOV     #SPBOT,SP       ;SET UP BOTTOM OF STACK,
002336  104011                          SRESET
002340  000167  177440                  JMP     GTRDYA
002344  011646                  EMTINT: MOV     @6,-(6)         ;GET SAVED PC,
002346  162716  000002                  SUB     #2,@6           ;DECREMENT PC BY 2,
002352  017616  000000                  MOV     @(6),@6
002356  121627  000022                  CMPB    @6,#22          ;CHECK THAT CALL IS
002362  101402                          BLOS    EMTA            ;WITHIN LIMITS,
002364  000000                          HALT                    ;CALL NOT WITHIN LIMITS,
002366  000776                          BR      .-2
002370  006116                  EMTA:   ROL     @6              ;EMT ARG X 2,
002372  042716  177001                  BIC     #177001,@6      ;REMOVE 7 MSB,
002376  062716  001272                  ADD     #EMTTAB,@6      ;FORM EMT RTN ADDR,
002402  017616  000000                  MOV     @(6),@6
002406  000136                          JMP     @(6)+           ;GO TO EMT ROUTINE,
002410  011646                  TRPINT: MOV     @6,-(6)         ;GET SAVED PC,
002412  162716  000002                  SUB     #2,@6           ;DECREMENT PC BY 2,
002416  017616  000000                  MOV     @(6),@6
002422  121627  000000                  CMPB    @6,#0           ;CHECK THAT EMT
002426  101402                          BLOS    TRPA            ;IS WITHIN LIMITS,
002430  000000                          HALT                    ;TRAP CALL NOT IN LIMIT,
002432  000776                          BR      .-2
```

```
002236  006110                  TRPA1:  ROL     @(6)            ;TRAP ARG X 2.
002240  042716  177001                  BIC     #177001,@(6)    ;REMOVE 7 MSB.
002244  062716  001340                  ADD     #TRPTAB,@(6)    ;FORM TRAP RTN ADDR.
002250  016616  000000                  MOV     @(6),@(6)
002254  000136                          JMP     @(6)+           ;GO TO TRAP ROUTINE.
002256  005767  176744          CHK33:  TST     TTYTYP          ;CHECK FOR 33.
002262  001002                          BNE     .+6             ;BRANCH IF NOT 33.
002264  062716  000002                  ADD     #2,@(6)         ;+2 TO EXIT POINTER
002270  000002                  CHK33B: RTI                     ;EXIT
002272  027767  000001  176726  CHK35:  CMP     #1,TTYTYP       ;CHECK FOR 35.
002300  001002                          BNE     .+6             ;BRANCH IF NOT 35.
002302  062716  000002                  ADD     #2,@(6)         ;+2 TO EXIT POINTER
002306  000002                          RTI                     ;EXIT
002310  032767  000010  176710  CKASR:  BIT     #BIT3,TTYTYP    ;CHECK FOR ASR TTY.
002316  001001                          BNE     .+4             ;BRANCH IF NOT ASR.
002320  000002                          RTI                     ;ASR. EXIT.
002322  022626                          POPSP2                  ;POP STACK TWICE.
002324  012767  000001  176710          MOV     #1,ICTR         ;FORCE I COUNT TO A 1.
002332  104012                          CHAIN                   ;CHAIN TO BYPASS ROUTINE.
                        ;SAVE REGS 0 TO 4 SUBROUTINE.
002334  012667  000030          SAVRG:  MOV     (6)+,SVRPC      ;SAVE PC AND PSW.
002340  012667  000026                  MOV     (6)+,SVRPSW
002344  010446                          MOV     %4,-(6)         ;SAVE REGS 0 - 4
002346  010346                          MOV     %3,-(6)         ;IN STACK.
002350  010246                          MOV     %2,-(6)
002352  010146                          MOV     %1,-(6)
002354  010046                          MOV     %0,-(6)
002356  016746  000010                  MOV     SVRPSW,-(6)     ;RESTORE PC AND PSW.
002362  016746  000002                  MOV     SVRPC,-(6)
002366  000002                          RTI                     ;EXIT.
002370  000000                  SVRPC:  OPEN
002372  000000                  SVRPSW: OPEN
                        ;RESTORE REGS 0 TO 4 SUBROUTINE.
002374  012667  000030          RSTRG:  MOV     (6)+,RSTPC      ;SAVE PC AND PSW.
002400  012667  000026                  MOV     (6)+,RSTPSW
002404  012600                          MOV     (6)+,%0         ;RESTORE REGS 0 - 4
002406  012601                          MOV     (6)+,%1         ;FROM STACK.
002410  012602                          MOV     (6)+,%2
002412  012603                          MOV     (6)+,%3
002414  012604                          MOV     (6)+,%4
002416  016746  000010                  MOV     RSTPSW,-(6)     ;RESTORE PC AND PSW.
002422  016746  000002                  MOV     RSTPC,-(6)
002426  000002                          RTI                     ;EXIT
002430  000000                  RSTPC:  OPEN
002432  000000                  RSTPSW: OPEN
```

```
                                ;ROUTINE TO FETCH A CHARACTER
002434  012767  000310  000308  AREAD:  MOV     #200.,BRCTR     ;SET UP DELAY COUNT.
002442  005277  176540          INC     @TKS            ;ENABLE READER.
002446  105777  176534  ARDA:   TSTB    @TKS            ;CHECK DONE BIT.
002452  100407                  BMI     ARDB            ;BRANCH IF DONE.
002454  104400                  DELAY                   ;DELAY 1 MILLISECOND.
002456  000001                  1
002460  005367  000256          DEC     BRCTR           ;TIME UP?
002464  001370                  BNE     ARDA            ;BRANCH IF TIME NOT UP YET.
002466  104010                  EMALT                   ;ERROR, NO RESPONSE FROM READER.
002470  000761                  BR      AREAD           ;TRY AGAIN.
002472  000207          ARDB:   RTS     %7              ;EXIT
                                ;ROUTINE TO SET LSR INTERRUPT VECTOR AND PRIORITY
002474  017667  000000  000012  STLSRV: MOV     @(6),STPRA+2    ;MOVE VECTOR ADDR TO STPRA+2
002502  062716  000002          ADD     #2,@%6          ;SET UP EXIT
002506  016701  176504          MOV     TKVTR,%1
002512  012721  000000  STPRA:  MOV     #OPEN,(1)+      ;SET VECTOR ADDRESS
002516  016721  176476          MOV     TKLVL,(1)+      ;SET PRIORITY
002522  000002                  RTI                     ;EXIT
                                ;ROUTINE TO SET LSP INTERRUPT VECTOR AND PRIORITY.
002524  017667  000000  000012  STLSPV: MOV     @(6),STPPA+2    ;MOVE VECTOR ADDR TO STPPA+2
002532  062716  000002          ADD     #2,@%6          ;SET UP EXIT
002536  016701  176460          MOV     TPVTR,%1
002542  012721  000000  STPPA:  MOV     #OPEN,(1)+      ;SET VECTOR ADDRESS.
002546  016721  176452          MOV     TPLVL,(1)+      ;SET PRIORITY
002552  000002                  RTI                     ;EXIT.
                                ;ROUTINE TO ISSUE RESET.
002554  012700  052525  SRSETT: MOV     #52525,%0               ;DATA TO R0.
002560  005100                  COM     %0              ;COMPLEMENT (R0).
002562  010067  177770          MOV     %0,SRSETT+2     ;(R0) TO SRSETT+2.
002566  000005                  RESET                   ;ISSUE RESET, (R0) IS
002570  000002                  RTI                     ;DISPLAYED, EXIT.
                                ;RANDOM NUMBER GENERATOR, ROUTINE EXITS WITH NUMBER IN REGISTER 0.
002572  016700  000042  RNGEN:  MOV     RP1,%0
002576  006100                  ROL     %0
002600  006100                  ROL     %0
002602  066700  000034          ADD     RP2,%0
002606  010067  000026          MOV     %0,RP1
002612  006100                  ROL     %0
002614  006100                  ROL     %0
002616  066700  000020          ADD     RP2,%0
002622  006100                  ROL     %0
002624  006100                  ROL     %0
002626  010067  000010          MOV     %0,RP2
002632  016700  000002          MOV     RP1,%0
002636  000207                  RTS     %7              ;EXIT, NUMBER IN R0
002640  001233          RP1:    1233
002642  007622          RP2:    7622
```

```
002044  104006                  BREAD:  STRDRV                  ;SET READER VECTOR
002046  002712                          BREADB                  ;TO BREADB
002050  052777  000101  176330          BIS     #101,@TKS       ;ENABLE LSR AND LSRI.
002056  012767  177777  000056          MOV     #177777,BRCTR   ;DELAY APPROX, 400 MSECS.
002064  005367  000052                  DEC     BRCTR
002070  001375                          BNE     .-4
002072  005077  176318                  CLR     @TKS            ;CLEAR LSRI ENABLE.
002076  104010                          EWALT                   ;NO RESPONSE HALT.
002100  000761                          BR      BREAD           ;TRY AGAIN.
002102  117767  176302  176434  BREADA: MOVB    @TKB,CRBUF      ;CHAR READ TO CRBUF.
002110  000207                          RTS     %7              ;EXIT SUBROUTINE.
002112  005077  176270          BREADB: CLR     @TKS            ;CLEAR LSR INTERRUPT ENABLE.
002116  105777  176264                  TSTB    @TKS            ;TEST FOR DONE.
002122  100003                          BPL     BREADC          ;BRANCH IF DONE NOT SET.
002124  012716  002102                  MOV     #BREADA,@X6     ;MODIFY INTERRUPT EXIT TO BREADA.
002130  000002                          RTI                     ;OK, EXIT INTERRUPT.
002132  000000          BREADC: HALT                            ;HALT, DONE BIT NOT SET AFTER INTERRUPT.
002134  012716  002044                  MOV     #BREAD,@X6      ;SET UP TO RETRY.
002140  000002                          RTI                     ;EXIT INTERRUPT.
002142  000000          BRCTR:  OPEN
                        ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
002144  011600          TYP:    MOV     @X6,X0                  ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
002146  062716  000002                  ADD     #2,@X6          ;SET UP EXIT.
002152  011000                          MOV     @X0,X0          ;ADDRESS OF MESSAGE TO R0.
002154  112067  000104          TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
002160  122767  000100  000076          CMPB    #100,TYPDAT     ;CHECK FOR"@"CHARACTER
002166  001001                          BNE     TYPC            ;BRANCH IF NOT"@".
002170  000002                          RTI                     ;TERMINATOR CHAR, DONE, EXIT.
002172  122767  000045  000064  TYPC:   CMPB    #45,TYPDAT      ;CHECK FOR"%".
003000  001416                          BEQ     TYPF            ;BRANCH IF"%"
003002  122767  000043  000054          CMPB    #43,TYPDAT      ;NOT"%",CHECK FOR"#".
003010  001417                          BEQ     TYPG            ;BRANCH IF "#"
003012  004767  000002                  JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
003016  000756                          BR      TYPA
003020  116777  000040  176166  TYPD:   MOVB    TYPDAT,@TPB     ;OUTPUT CHARACTER TO PRINTER
003026  105777  176160                  TSTB    @TPS            ;WAIT FOR DONE FLAG.
003032  100375                          BPL     .-4
003034  000207                          RTS     %7              ;EXIT
003036  112767  000015  000020  TYPF:   MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
003044  004767  177750                  JSR     %7,TYPD         ;GO TYPE CHAR.
003050  112767  000012  000006  TYPG:   MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT.
003056  004767  177736                  JSR     %7,TYPD         ;GO TYPE CHAR.
003062  000734                          BR      TYPA
003064  000000          TYPDAT: OPEN
                        ;SUBROUTINE TO OUTPUT A SERIES OF ASCII MESSAGES ON TELETYPE PRINTER
003066  011600          TYPS:   MOV     @X0,X0                  ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
003070  062716  000002                  ADD     #2,@X6          ;UPDATE TO NEXT MESSAGE ADDRESS
003074  011067  000014                  MOV     @X0,TYPSB       ;ADDRESS OF MESSAGE TO TYPSB
003100  022767  177777  000006          CMP     #-1,TYPSB       ;CHECK FOR TERMINATOR
003106  001001                          BNE     TYPSA           ;BRANCH IF NOT TERMINATOR.
003110  000002                          RTI                     ;TERMINATOR, EXIT
003112  104000          TYPSA:  TYPE                            ;CALL ON TYP SUB TO TYPE MESSAGE
003114  000000          TYPSB:  OPEN                            ;ADDRESS OF MESSAGE GOES HERE
003116  000763                  BR      TYPS                    ;GO PROCESS NEXT MESSAGE
```

```
                                        ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
003120  011667  000036          DLY:    MOV     @X6,DLCNT       ;GET DELAY COUNT ADDRESS,
003124  062716  000002                  ADD     #2,@X6          ;SET UP EXIT ADDRESS
003130  017746  000026                  MOV     @DLCNT,-(6)     ;DELAY COUNT TO STACK
003134  005067  174636                  CLR     PSW             ;SET PRIORITY 0
003140  012746  000226          DLYA:   MOV     #226,-(6)       ;1 MSEC COUNT TO STACK
003144  005316                  DLYB:   DEC     @X6             ;DECREMENT 1 MSEC COUNT
003146  001376                          BNE     DLYB            ;BRANCH IF NOT 0,
003150  005726                          POPSP                   ;ZERO, UNCOVER MSECS, COUNT,
003152  005316                          DEC     @X6             ;DECREMENT IT
003154  001371                          BNE     DLYA            ;BR IF NOT DONE DELAYING
003156  005726                          POPSP                   ;DONE
003160  000002                          RTI                     ;EXIT,
003162  000000                  DLCNT:  OPEN                    ;CONTAINS MILLISECONDS COUNT ADDRESS,
                                        ;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS, MAXIMUM STALL
                                        ;DETERMINED BY CONTENTS OF LOC STLMSK,
003164  032767  040000  176054  STAL:   BIT     #BIT14,PRGID    ;TEST FOR STALLS ALLOWED,
003172  001001                          BNE     STALAA          ;ALLOWED,
003174  000002                          RTI                     ;NOT ALLOWED,
003176  004767  177370          STALAA: JSR     X7,RNGEN        ;GO GET RANDOM NUMBER,
003202  046700  000014                  BIC     STLMSK,X0       ;0 IN R0, APPLY STALL MASK,
003206  001404                          BEQ     STALB           ;BRANCH IF RESULT IS 0,
003210  010067  000002                  MOV     X0,STALA
003214  104400                          DELAY                   ;DELAY
003216  000000                  STALA:  OPEN                    ;DELAY COUNT
003220  000002                  STALB:  RTI                     ;DONE, EXIT,
003222  000000                  STLMSK: OPEN                    ;STALL MASK,
                                        ;SUBROUTINE TO GENERATE RANDOM CHARACTER COUNT
003224  004767  177342          GRCNT:  JSR     X7,RNGEN        ;GET RANDOM NUMBER
003230  046700  000010                  BIC     RCMSK,X0        ;APPLY MASK
003234  001773                          BEQ     GRCNT           ;TRY AGAIN IF RESULT 0
003236  010067  000004                  MOV     X0,RNCNT        ;COUNT TO RNCNT
003242  000207                          RTS     X7              ;EXIT,
003244  000000                  RCMSK:  OPEN                    ;RANDOM CHARACTER MASK,
003246  000000                  RNCNT:  OPEN                    ;RANDOM CHARACTER COUNT,
                                        ;SUBROUTINE TO COMPARE DATA READ FROM READER AGAINST EXPECTED DATA AND REPORT ERRORS,
003250  004767  000260          BCHECK: JSR     X7,GTBIN        ;GET BIN CHARACTER(IN R0)
003254  110067  176065                  MOVB    X0,CRBUF+1      ;S/B CHAR TO CRBUF+1
003260  126767  176060  176057          CMPB    CRBUF,CRBUF+1   ;COMPARE S/B AND WAS CHARS,
003266  001001                          BNE     .+4             ;BRANCH IF NOT SAME,
003270  000207                          RTS     X7              ;SAME, EXIT,
003272  104017                          DATHLT                  ;GO HALT AND DISPLAY DATA,
003274  005367  176054                  DEC ERCTR               ;3 ERRORS?
003300  001002                          BNE     .+6             ;BRANCH IF NOT 3 YET,
003302  004767  000002                  JSR     X7,BSYNC        ;3 ERRORS, RESYNC READER,
003306  000207                          RTS     X7              ;EXIT,
```

```
                                ;SUBROUTINE TO SYNC THE LSR TO A SPECIAL BINARY COUNT PATTERN TEST TAPE.
003310  004767  000162          BSYNC:  JSR     %7,INBIN        ;INITIALIZE BINARY PATTERN
003314  004767  177324                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR1
003320  116767  176020  176020          MOVB    CRBUF,CHR1
003326  004767  177312                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR2
003332  116767  176006  176010          MOVB    CRBUF,CHR2
003340  004767  177300                  JSR     %7,BREAD        ;READ CHAR AND STORE AT CHR3.
003344  116767  175774  176000          MOVB    CRBUF,CHR3
003352  004767  000012                  JSR     %7,SYNCA        ;GO SYNC
003356  000754                          BR      BSYNC           ;NO SYNC, TRY AGAIN.
003360  012767  000003  175766          MOV     #3,ERCTR
003366  000207                          RTS     %7              ;SUCCESS,EXIT.
003370  012767  001000  000074  SYNCA:  MOV     #512.,SYCTRA    ;512 TO SYCTRA
003376  012767  000012  000070          MOV     #10.,SYCTRB     ;10 TO SYCTRB
003404  004767  000124  SYNCB:  JSR     %7,GTBIN        ;GET BIN CHARACTER(CHAR IN R0)
003410  120067  175732                  CMPB    %0,CHR1         ;COMPARE TO CHR1
003414  001373                          BNE     SYNCB           ;BRANCH IF NOT EQUAL
003416  004767  000112                  JSR     %7,GTBIN        ;SAME, GET ANOTHER BIN CHAR.
003422  120067  175722                  CMPB    %0,CHR2         ;COMPARE TO CHR2
003426  001405                          BEQ     SYNCD           ;BRANCH IF EQUAL
003430  005367  000036                  DEC     SYCTRA          ;DECREMENT SYCTRA
003434  001363                          BNE     SYNCB           ;BRANCH IF NOT DONE 512 TIMES.
003436  104010  SYNCC:  EMALT                                   ;DONE 512, SYNC ERROR.
003440  000207                          RTS     %7              ;ERROR EXIT
003442  004767  000066  SYNCD:  JSR     %7,GTBIN        ;GET BIN CHARACTER
003446  120067  175700                  CMPB    %0,CHR3         ;COMPARE TO CHR3
003452  001404                          BEQ     SYNCE           ;BRANCH IF SAME
003454  005367  000014                  DEC     SYCTRB          ;DECREMENT SYCTRB.
003460  001351                          BNE     SYNCB           ;BRANCH IF NOT DONE 10 TIMES
003462  000765                          BR      SYNCC           ;SYNC ERROR, BRANCH
003464  062716  000002  SYNCE:  ADD     #2,0%6          ;SET UP SUCCESS EXIT
003470  000207                          RTS     %7              ;EXIT.
003472  000000  SYCTRA:  OPEN
003474  000000  SYCTRB:  OPEN
                                ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
003476  012767  177777  000014  INBIN:  MOV     #-1,RIND        ;SET ALL VARIABLES
003504  004567  000300                  JSR     %5,BMOVE        ;TO MINUS 1.
003510  003520                          RIND
003512  003521                          RIND+1
003514  000013                          11.
003516  000207                          RTS     %7              ;EXIT
003520  000000  RIND:   OPEN
003522  000000  PT0:    OPEN
003524  000000  PT1:    OPEN
003526  000000  PIND:   OPEN
003530  000000  PT0P:   OPEN
003532  000000  PT1P:   OPEN
```

```
                                        ;SPECIAL BINARY COUNT PATTERN SUBROUTINE, EXITS WITH BIN CHAR IN R0
003534  016767  177762  177762  GTBIN:  MOV     PT0,PT1         ;PREVIOUS BIN CHAR TO PT1
003542  005167  177756          GTBIN1: COM     PT1
003546  005167  177746          GTBIN2: COM     RIND
003552  001002                  GTBIN3: BNE     .+6
003554  005267  177744                  INC     PT1
003560  042767  177400  177736          BIC     #177400,PT1     ;MASK TO 8 BITS
003566  016767  177732  177726          MOV     PT1,PT0         ;SAVE BIN CHAR IN PT0
003574  016700  177724                  MOV     PT1,%0          ;BIN CHAR TO R0,
003600  000207                          RTS     %7              ;EXIT,
003602  016767  177722  177722  GTBINP: MOV     PT0P,PT1P       ;PREVIOUS BIN CHAR TO PT1P
003610  005167  177716          GTBIN4: COM     PT1P
003614  005167  177706          GTBIN5: COM     PIND
003620  001002                  GTBIN6: BNE     .+6
003622  005267  177704                  INC     PT1P
003626  042767  177400  177676          BIC     #177400,PT1P    ;MASK TO 8 BITS,
003634  016767  177672  177666          MOV     PT1P,PT0P       ;SAVE BIN CHAR IN PT0P,
003642  016701  177664                  MOV     PT1P,%1         ;BIN CHAR TO R1,
003646  000207                          RTS     %7              ;EXIT,
                                        ;OCTAL TO ASCII CONVERT ROUTINES
003650  012500                  ACNV6:  MOV     (5)+,%0         ;CONVERT TO 6 ASCII, GET OCTAL ADDRESS
003652  012567  000012                  MOV     (5)+,ACNVB      ;GET ASCII ADDRESS
003656  004767  000052                  JSR     %7,ACNV         ;CONVERT TO ASCII
003662  004567  000122                  JSR     %5,BMOVE        ;MOVE 6 CHARS TO ASCII ADDRESS
003666  003724                          .-A1ST
003670  000000                  ACNVB:  OPEN
003672  000006                          6
003674  000205                          RTS     %5              ;EXIT
003676  012500                  ACNV4:  MOV     (5)+,%0         ;CONVERT TO 4 ASCII, GET OCTAL ADDRESS
003700  012567  000012                  MOV     (5)+,ACNVC      ;GET ASCII ADDRESS
003704  004767  000024                  JSR     %7,ACNV         ;CONVERT TO ASCII
003710  004567  000074                  JSR     %5,BMOVE        ;MOVE 4 CHARS TO ASCII ADDRESS',
003714  003726                          .-A1ST+2
003716  000000                  ACNVC:  OPEN
003720  000004                          4
003722  000205                          RTS     %5              ;EXIT
003724  000000                  A1ST:   OPEN
003726  000000                          OPEN
003730  000000                          OPEN
003732  000000                  ACNVX:  OPEN
003734  012701  003732          ACNV:   MOV     #A1ST+6,%1      ;ADDR TO STORE ASCII TO R1
003740  012702  000006                  MOV     #6,%2           ;6 TO R2
003744  011067  177762                  MOV     @%0,ACNVX       ;OCTAL WORD TO ACNVX
003750  016703  177756          ACNVM:  MOV     ACNVX,%3
003754  042703  177770                  BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT OCTAL #
003760  062703  000060                  ADD     #60,%3          ;ADD 60 TO CONVERT TO ASCII
003764  110341                          MOVB    %3,-(1)         ;STORE ASCII BYTE
003766  006067  177740                  ROR     ACNVX           ;MOVE NEXT OCTAL DIGIT TO LEAST
003772  006067  177734                  ROR     ACNVX           ;SIGNIFICANT POSITION
003776  006067  177730                  ROR     ACNVX
004002  005302                          DEC     %2              ;DONE 6 TIMES?
004004  001361                          BNE     ACNVM           ;NO, REPEAT,
004006  000207                          RTS     %7              ;YES, EXIT,
```

```
                                ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
004010  104020                  BMOVE:  SAVREG                  ;SAVE REGS.
004012  012501                          MOV     (5)+,R1         ;GET "FROM" ADDRESS
004014  012502                          MOV     (5)+,R2         ;GET "TO" ADDRESS
004016  012503                          MOV     (5)+,R3         ;GET COUNT
004020  112122                  BMOVA:  MOVB    (1)+,(2)+       ;MOVE BYTE
004022  005303                          DEC     R3              ;DECREMENT COUNT
004024  001375                          BNE     BMOVA           ;BRANCH IF NOT DONE.
004026  104021                          RSTREG                  ;RESTORE REGS.
004030  000205                          RTS     R5              ;DONE EXIT
                                ;SUBROUTINE TO CHECK FOR PUNCH READY.
004032  105777  175154          CPRDY:  TSTB    @TPS            ;TEST FOR READY BIT.
004036  100001                          BPL     CPRDYA          ;BRANCH IF READY NOT SET.
004040  000207                          RTS     R7              ;OK, EXIT.
004042  104010                  CPRDYA: EMALT                   ;NOT READY, HALT.
004044  000772                          BR      CPRDY
                                ;SUBROUTINE TO PUNCH ON LSP CHARACTER IN REG 0.
004046  004767  177760          LSPCH:  JSR     R7,CPRDY        ;GO CHECK FOR PUNCH READY.
004052  010077  175136                  MOV     R0,@TPB         ;LOAD PUNCH BUFFER.
004056  105777  175130                  TSTB    @TPS            ;WAIT FOR DONE.
004062  100375                          BPL     .-4
004064  005000                          CLR     R0
004066  000207                          RTS     R7              ;DONE, EXIT.
                                ;BINARY TO DECIMAL ASCII CONVERT SUBROUTINE.
004070  012700  011664          BDCNV:  MOV     #DECVAL,R0      ;SET UP ADDR TO STORE DECIMAL ASCII IN R2
004074  013501                          MOV     @(5)+,R1        ;BINARY VALUE TO R1.
004076  012702  004176                  MOV     #ADTENP,R2      ;ADDR OF TEN POWER STRING TO R2.
004102  012767  000005  000060          MOV     #5,CNVCTR       ;SET UP FOR 5 POWER CONVERSIONS.
004110  012267  000060          BDCNVA: MOV     (2)+,TENPWR     ;MOVE POWER OF TEN VALUE TO TENPWR.
004114  004767  000010                  JSR     R7,SUBTEN       ;PERFORM CONVERSION
004120  005367  000044                  DEC     CNVCTR          ;DONE 5 CONVERSIONS?
004124  001371                          BNE     BDCNVA          ;BRANCH IF NOT YET 5.
004126  000205                          RTS     R5              ;YES, EXIT.
004130  005067  000036          SUBTEN: CLR     DIGIT           ;CLEAR DIGIT
004134  166701  000034          SUBTNA: SUB     TENPWR,R1       ;SUBTRACT TEN POWER FROM BINARY VALUE.
004140  103403                          BCS     SUBTNB          ;BRANCH IF UNSUCCESSFUL SUBTRACTION.
004142  005267  000024                  INC     DIGIT
004146  000772                          BR      SUBTNA
004150  066701  000020          SUBTNB: ADD     TENPWR,R1       ;RESTORE SUBTRACTED VALUE.
004154  062767  000060  000010          ADD     #60,DIGIT       ;CONVERT (DIGIT) TO ASCII
004162  116720  000004                  MOVB    DIGIT,(0)+      ;MOVE ASCII CHAR TO DECVAL FIELD.
004166  000207                          RTS     R7              ;EXIT.
004170  000000          CNVCTR: OPEN
004172  000000          DIGIT:  OPEN
004174  000000          TENPWR: OPEN
004176  023420          ADTENP: 10000.
004200  001750                  1000.
004202  000144                  100.
004204  000012                  10.
004206  000001                  1
```

```
                                ;SUBROUTINE TO TYPE A LINE OF CHARACTERS
004210  012767  000112  000024  TYPLN:  MOV     #74.,,TCTR      ;72 TO CHAR COUNT +CR,LF
004216  012704  011671          TYPLA:  MOV     #BLOCKA,X4      ;SET LINE ADDRESS IN R4.
004222  104002                  TYPLB:  STALL                   ;STALL IF ALLOWED.
004224  112400                          MOVB    (4)+,X0         ;GET CHARACTER
004226  004767  177614                  JSR     X7,LSPCH        ;GO OUTPUT CHARACTER.
004232  005367  000004                  DEC     TCTR            ;DONE?
004236  001371                          BNE     TYPLB           ;BRANCH IF NOT DONE.
004240  000207                          RTS     X7              ;DONE, EXIT
004242  000000          TCTR:   OPEN
                                ;SUBROUTINE TO TYPE LINE OF 3 CHARACTERS
004244  011667  000016          TYPL3:  MOV     @X6,TPL3A       ;DEVELOP AND SET ADDRESS OF
004250  017767  000012  000010          MOV     @TPL3A,TPL3A    ;DATA IN TPL3A.
004256  062716  000002                  ADD     #2,@X6          ;SET UP EXIT.
004262  004567  000034                  JSR     X5,FBF3         ;FILL BUFFER WITH 3 CHARACTERS
004266  000000          TPL3A:  OPEN
004270  042767  040000  174758          BIC     #BIT14,PRGID    ;DISABLE STALLS.
004276  004767  177706                  JSR     X7,TYPLN        ;GO TYPE LINE OF CHARACTERS.
004302  000002                          RTI                     ;EXIT.
004304  112767  000015  005357  STBF:   MOVB    #15,BLOCKA      ;SUB TO SET UP BUFFER AREA.
004312  112767  000012  005352          MOVB    #12,BLOCKA+1
004320  000207                          RTS     X7              ;EXIT
                                ;SUBROUTINE TO FILL CHARACTER BUFFER WITH 3 CHARACTERS.
004322  012567  000004  FBF3:   MOV     (5)+,FBF3A
004326  004567  177456          FBF3A:  JSR     X5,BMOVE        ;MOVE 3 CHARS TO BUFFER.
004332  000000          FBF3A1: OPEN
004334  011673                  BLOCK1
004336  000003                  3
004340  004567  177444  FBF3B:  JSR     X5,BMOVE        ;FILL 72 CHARACTERS BUFFER
004344  011673                  BLOCK1                  ;WITH 3 CHARACTERS
004346  011676                  BLOCK1+3
004350  000105                  69.
004352  004567  177432          JSR     X5,BMOVE
004356  011673                  BLOCK1
004360  012005                  BLOCK2
004362  000110                  72.
004364  000205                  RTS     X5              ;EXIT
```

```
                                ;SUBROUTINE TO FILL BUFFER WITH ALL CHARACTERS
004366  004567  177416  FBALL1  JSR     X5,BMOVE        ;FILL 72 CHAR BUFFER WITH
004372  010714                  A                       ;ALL CHARACTERS,
004374  011673                  BLOCK1
004376  000077                  63.
004400  004567  177404          JSR     X5,BMOVE
004404  010714                  A
004406  011772                  BLOCK1+63.
004410  000011                  9.
004412  004567  177372          JSR     X5,BMOVE
004416  011673                  BLOCK1
004420  012005                  BLOCK2
004422  000110                  72.
004424  000207                  RTS     X7              ;EXIT.
                                ;SUB TO FILL BUFFER WITH 33 WORST CASE PATTERN,
004426  004567  177356  FW3361  JSR     X5,BMOVE        ;16 CHARACTER PATTERN TO BUFFER
004432  010700                  A33WP6
004434  011673                  BLOCK1
004436  000006                  6
004440  004567  177344          JSR     X5,BMOVE        ;FILL BUFFER WITH PATTERN,
004444  011673                  BLOCK1
004446  011701                  BLOCK1+6
004450  000102                  66.
004452  004567  177332          JSR     X5,BMOVE
004456  011673                  BLOCK1
004460  012005                  BLOCK2
004462  000110                  72.
004464  000207                  RTS     X7              ;EXIT

                                ;SUB TO FILL BUFFER WITH 35 WORST CASE PATTERN,
004466  004567  177316  FW3561  JSR     X5,BMOVE        ;16 CHARACTER PATTERN TO BUFFER
004472  010706                  A35WP6
004474  011673                  BLOCK1
004476  000006                  6
004500  004567  177304          JSR     X5,BMOVE        ;FILL BUFFER WITH PATTERN,
004504  011673                  BLOCK1
004506  011701                  BLOCK1+6
004510  000102                  66.
004512  004567  177272          JSR     X5,BMOVE
004516  011673                  BLOCK1
004520  012005                  BLOCK2
004522  000110                  72.
004524  000207                  RTS     X7              ;EXIT.
```

```
                                ;PRGB - INPUT-OUTPUT LOGIC TESTS
004526  012767  004540  174476  PRGB:   MOV     #AT0,KSTART     ;ADDRESS OF 1ST ROUTINE TO KSTART,
004534  000167  175032          PRGB1:  JMP     SRSET           ;GO GET STARTED,
                                ;TEST ABILITY TO REFERENCE THE KEYBOARD/READER STATUS WORD (TKS)
004540  000000                  AT0:    0                       ;TEST #,
004542  004572                          AT1                     ;NEXT TEST,
004544  001750                          1000.                   ;1 COUNT,
004546  004556                          AT0A                    ;SCOPE ENTRY,
004550  012767  004566  173226          MOV     #AT0E,MACHER    ;SET UP MACHINE ERROR TRAP,
004556  005777  174424          AT0A:   TST     @TKS            ;REFERENCE CODER STATUS WORD,
004562  104012                  AT0B:   CHAIN                   ;CHAIN
004564  000774                          BR      AT0A            ;REPEAT TEST,
004566  104003                  AT0E:   ERROR                   ;ERROR, TRAPPED WHEN REFERENCING READER,
004570  000774                          BR      AT0B            ;STATUS WORD (TKS),
                                ;TEST ABILITY TO REFERENCE THE KEYBOARD/READER BUFFER (TKB),
004572  000001                  AT1:    1                       ;TEST #,
004574  004624                          AT2                     ;NEXT TEST,
004576  001750                          1000.                   ;1 COUNT,
004600  004610                          AT1A                    ;SCOPE ENTRY,
004602  012767  004620  173174          MOV     #AT1E,MACHER    ;SET UP MACHINE ERROR TRAP
004610  005777  174374          AT1A:   TST     @TKB            ;REFERENCE READER BUFFER,
004614  104012                  AT1B:   CHAIN                   ;CHAIN
004616  000774                          BR      AT1A            ;REPEAT TEST,
004620  104003                  AT1E:   ERROR                   ;ERROR, TRAPPED WHEN REFERENCING
004622  000774                          BR      AT1B            ;READER BUFFER, (TKB),
                                ;TEST ABILITY TO REFERENCE PUNCH/PRINTER STATUS WORD (TPS),
004624  000002                  AT2:    2                       ;TEST #,
004626  004656                          AT3                     ;NEXT TEST
004630  001750                          1000.                   ;1 COUNT,
004632  004642                          AT2A                    ;SCOPE ENTRY,
004634  012767  004652  173142          MOV     #AT2E,MACHER    ;SETUP MACHINE ERROR TRAP,
004642  005777  174344          AT2A:   TST     @TPS            ;REFERENCE PUNCH/PRINTER STATUS WORD,
004646  104012                  AT2B:   CHAIN                   ;CHAIN
004650  000774                          BR      AT2A            ;REPEAT TEST,
004652  104003                  AT2E:   ERROR                   ;ERROR, TRAPPED WHEN REFERENCING
004654  000774                          BR      AT2B            ;PUNCH/PRINTER STATUS WORD (TPS),
                                ;TEST ABILITY TO REFERENCE PUNCH/PRINTER BUFFER (TPB),
004656  000003                  AT3:    3
004660  004710                          AT4
004662  001750                          1000.
004664  004674                          AT3A
004666  012767  004704  173110          MOV     #AT3E,MACHER    ;SETUP MACHINE ERROR TRAP,
004674  005777  174314          AT3A:   TST     @TPB            ;REFERENCE PUNCH/PRINTER BUFFER,
004700  104012                  AT3B:   CHAIN                   ;CHAIN
004702  000774                          BR      AT3A            ;REPEAT TEST,
004704  104003                  AT3E:   ERROR                   ;ERROR, TRAPPED WHEN REFERENCING
004706  000774                          BR      AT3B            ;PUNCH/PRINTER BUFFER, (TPS),
```

```
                                ;TEST ABILITY TO SET AND CLEAR READER/KYBD ID BIT
004710  000004                  AT4:    4                       ;TEST #
004712  004774                          AT5                     ;NEXT TEST
004714  001750                          1000.                   ;I COUNT
004716  004726                          AT4A                    ;SCOPE ENTRY
004720  012767  000340  173050          MOV     #PRTY7,PSW      ;SET PRIORITY 7.
004726  052777  000100  174252  AT4A1:  BIS     #BIT6,@TKS      ;SET ID BIT IN TKS.
004734  032777  000100  174244          BIT     #BIT6,@TKS      ;CHECK ID BIT IN TKS
004742  001002                          BNE     AT4B            ;BRANCH IF ID BIT IS SET.
004744  104003                  AT4E1:  ERROR                   ;ERROR 1 ID BIT NOT SET.
004746  000410                          BR      AT4C
004750  042777  000100  174230  AT4B:   BIC     #BIT6,@TKS      ;CLEAR ID BIT IN TKS
004756  032777  000100  174222          BIT     #BIT6,@TKS      ;CHECK ID BIT IN TKS.
004764  001401                          BEQ     AT4C            ;BRANCH IF ID BIT IS CLEARED.
004766  104003                  AT4E2:  ERROR                   ;ERROR, ID BIT FAILED TO CLEAR.
004770  104012                  AT4C:   CHAIN                   ;CHAIN
004772  000755                          BR      AT4A            ;REPEAT TEST.
                                ;TEST ABILITY TO CLEAR ID BIT WITH RESET INSTRUCTION.
004774  000005                  AT5:    5                       ;TEST #
004776  005040                          AT24                    ;NEXT TEST
005000  000144                          100.                    ;I COUNT
005002  005012                          AT5A                    ;SCOPE ENTRY.
005004  012767  000340  172764          MOV     #PRTY7,PSW      ;SET PRIORITY 7.
005012  052777  000100  174166  AT5A:   BIS     #BIT6,@TKS      ;SET ID BIT IN TKS
005020  104011                          SRESET                  ;RESET
005022  032777  000100  174156          BIT     #BIT6,@TKS      ;TEST ID BIT.
005030  001401                          BEQ     AT5B            ;BRANCH IF ID BIT IS CLEAR.
005032  104003                  AT5E:   ERROR                   ;ERROR, RESET FAILED TO CLEAR ID BIT.
005034  104012                  AT5B:   CHAIN                   ;CHAIN
005036  000765                          BR      AT5A            ;REPEAT TEST.
```

```
                                        ;TEST ABILITY TO SET AND CLEAR PUNCH ID BIT
005040  000024                  AT24:   24                      ;TEST#
005042  005124                          AT25                    ;NEXT TEST,
005044  001750                          1000.                   ;1 COUNT
005046  005056                          AT24A                   ;SCOPE ENTRY,
005050  012767  000340  172720          MOV     #PRTY7,PSW      ;SET PRIORITY 7,
005056  052777  000100  174126  AT24A1: BIS     #BIT6,@TPS      ;SET PUNCH ID BIT,
005064  032777  000100  174120          BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT,
005072  001002                          BNE     AT24B           ;BRANCH IF PUNCH ID BIT IS SET,
005074  104003                  AT24E1: ERROR                   ;ERROR1, PUNCH ID BIT DID NOT SET,
005076  000410                          BR      AT24C
005100  042777  000100  174104  AT24B1: BIC     #BIT6,@TPS      ;CLEAR PUNCH ID BIT,
005106  032777  000100  174076          BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT,
005114  001401                          BEQ     AT24C           ;BRANCH IF PUNCH ID BIT IS CLEAR
005116  104003                  AT24E2: ERROR                   ;ERROR2, PUNCH ID BIT FAILED TO CLEAR,
005120  104012                  AT24C:  CHAIN                   ;CHAIN
005122  000776                          BR      AT24C           ;REPEAT TEST
                                        ;TEST ABILITY TO CLEAR PUNCH ID BIT WITH RESET INSTRUCTION
005124  000025                  AT25:   25                      ;TEST#
005126  005170                          AT26                    ;NEXT TEST,
005130  000144                          100.                    ;1 COUNT,
005132  005142                          AT25A                   ;SCOPE ENTRY,
005134  012767  000340  172634          MOV     #PRTY7,PSW      ;SET PRIORITY 7,

005142  052777  000100  174042  AT25A1: BIS     #BIT6,@TPS      ;SET PUNCH ID BIT,
005150  104011                          SRESET                  ;RESET
005152  032777  000100  174032          BIT     #BIT6,@TPS      ;CHECK PUNCH ID BIT,
005160  001401                          BEQ     AT25B           ;BRANCH IF PUNCH ID BIT IS CLEAR,
005162  104003                  AT25E1: ERROR                   ;ERROR, RESET FAILED TO CLEAR PUNCH ID BIT,
005164  104012                  AT25B1: CHAIN                   ;CHAIN
005166  000765                          BR      AT25A           ;REPEAT TEST,
                                        ;TEST ABILITY TO SET AND CLEAR THE PUNCH MAINTENANCE BIT
005170  000026                  AT26:   26                      ;TEST#
005172  005246                          AT27                    ;NEXT TEST
005174  001750                          1000.                   ;1 COUNT
005176  005200                          AT26A                   ;SCOPE ENTRY
005200  052777  000004  174004  AT26A1: BIS     #BIT2,@TPS      ;SET MAINTANCE BIT,
005206  032777  000004  173776          BIT     #BIT2,@TPS      ;CHECK MAINTENANCE BIT
005214  001002                          BNE     AT26B           ;BRANCH IF MAINTENANCE BIT SET,
005216  104003                  AT26E1: ERROR                   ;ERROR1, MAINTENANCE BIT FAILED TO SET,
005220  000410                          BR      AT26C
005222  042777  000004  173762  AT26B1: BIC     #BIT2,@TPS      ;CLEAR MAINTENANCE BIT,
005230  032777  000004  173754          BIT     #BIT2,@TPS      ;CHECK MAINTENANCE BIT
005236  001401                          BEQ     AT26C           ;BRANCH IF MAINTENANCE BIT IS CLEAR,
005240  104003                  AT26E2: ERROR                   ;ERROR2, MAINTENANCE BIT FAILED TO CLEAR,
005242  104012                  AT26C1: CHAIN                   ;CHAIN
005244  000755                          BR      AT26A           ;REPEAT TEST
```

```
                                ;TEST THAT RESET INSTRUCTION CLEARS THE MAINTENANCE BIT.
005246  000027                  AT27:   27                      ;TEST#
005250  005304                          AT30                    ;NEXT TEST
005252  000144                          100.                    ;I COUNT
005254  005256                          AT27A                   ;SCOPE ENTRY
005256  052777  000004  173726  AT27A:  BIS     #BIT2,@TPS      ;SET MAINTENANCE BIT.
005264  104411                          SRESET                  ;ISSUE RESET
005266  032777  000004  173716          BIT     #BIT2,@TPS      ;CHECK MAINTENANCE BIT
005274  001401                          BEQ     AT27B           ;BRANCH IF MAINTENANCE BIT CLEAR.
005276  104003                  AT27E:  ERROR                   ;ERROR, RESET FAILED TO CLEAR
005300  104012                  AT27B:  CHAIN                   ;THE MAINTENANCE BIT, CHAIN.
005302  000765                          BR      AT27A           ;REPEAT TEST.
                                ;TEST THAT RESET SETS THE PUNCH READY BIT, AND THAT READY CAN BE READ RELIABLY.
005304  000030                  AT30:   30                      ;TEST#
005306  005330                          AT31                    ;NEXT TEST
005310  001750                          1000.                   ;I COUNT
005312  005314                          AT30A                   ;SCOPE ENTRY
005314  105777  173672          AT30A:  TSTB    @TPS            ;CHECK PUNCH READY.
005320  100401                          BMI     AT30B           ;BRANCH IF PUNCH READY IS SET.
005322  104003                  AT30E:  ERROR                   ;ERROR, RESET FAILED TO SET READY, OR FAILED TO READ  IT.
005324  104012                  AT30B:  CHAIN                   ;CHAIN
005326  000772                          BR      AT30A           ;REPEAT TEST.
                                ;TEST THAT PUNCH READY RESETS BY LOADING PUNCH BUFFER.
005330  000031                  AT31:   31                      ;TEST#
005332  005366                          AT32                    ;NEXT TEST
005334  000024                          20.                     ;I COUNT
005336  005340                          AT31A                   ;SCOPE ENTRY
005340  104400                  AT31A:  DELAY                   ;WAIT 150 MSECS
005342  000226                          150.
005344  104011                          SRESET                  ;RESET
005346  005077  173642                  CLR     @TPB            ;LOAD PUNCH BUFFER
005352  105777  173634                  TSTB    @TPS            ;CHECK PUNCH READY BIT.
005356  100001                          BPL     AT31B           ;BRANCH IF PUNCH READY IS CLEAR.

005360  104003                  AT31E:  ERROR                   ;ERROR, BUFFER LOAD FAILED TO CLEAR READY.
005362  104012                  AT31B:  CHAIN                   ;CHAIN
005364  000765                          BR      AT31A           ;REPEAT TEST.
                                ;TEST THAT BYTE LOAD OF PUNCH BUFFER +1 DOES NOT RESET READY.
005366  000032                  AT32:   32                      ;TEST#
005370  005430                          AT33                    ;NEXT TEST
005372  000024                          20.                     ;I COUNT
005374  005376                          AT32A                   ;SCOPE ENTRY
005376  104400                  AT32A:  DELAY                   ;WAIT 150 MSECS
005400  000226                          150.
005402  104011                          SRESET                  ;RESET
005404  016700  173604                  MOV     TPB,R0
005410  005200                          INC     R0
005412  105010                          CLRB    @R0             ;BYTE LOAD PUNCH BUFFER+1
005414  105777  173572                  TSTB    @TPS            ;CHECK PUNCH READY BIT
005420  100401                          BMI     AT32B           ;BRANCH IF PUNCH READY STILL SET.
005422  104003                  AT32E:  ERROR                   ;ERROR, BYTE LOAD OF PUNCH BUFFER+1
005424  104012                  AT32B:  CHAIN                   ;CLEARED READY, CHAIN
005426  000763                          BR      AT32A           ;REPEAT TEST.
```

```
                                ;TEST THAT PUNCH BECOMES READY BY 200 MSECS AFTER BUFFER LOAD.
005430  000033                  AT33:   33                      ;TEST #
005432  005470                          AT34                    ;NEXT TEST
005434  000024                          20.                     ;I COUNT
005436  005440                          AT33A                   ;SCOPE ENTRY.
005440  104400                  AT33A:  DELAY                   ;WAIT 150 MSECS.
005442  000226                          150.
005444  005077  173544                  CLR     @TPB            ;LOAD PUNCH BUFFER.
005450  104400                          DELAY                   ;WAIT 200 MSECS.
005452  000310                          200.
005454  105777  173532                  TSTB    @TPS            ;CHECK PUNCH READY BIT.
005460  100401                          BMI     AT33B           ;BRANCH IF PUNCH READY IS SET.
005462  104003                  AT33E:  ERROR                   ;ERROR, READY NOT SET 200 MSECS AFTER BUFFER LOAD.
005464  104012                  AT33B:  CHAIN                   ;CHAIN
005466  000764                          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.
005470  000034                  AT34:   34                      ;TEST #
005472  005536                          AT35                    ;NEXT TEST
005474  001750                          1000.                   ;I COUNT
005476  005504                          AT34A                   ;SCOPE ENTRY
005500  104007                          STPCHV                  ;SET PUNCH INTERRUPT SERVICE
005502  005532                          AT34C                   ;TO AT34C
005504  005077  173502          AT34A:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
005510  005067  172262                  CLR     PSW             ;SET PRIORITY 0.
005514  052777  000100  173470          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS.
005522  000240                          NOP
005524  104003                  AT34E:  ERROR                   ;PUNCH READY FAILED TO CAUSE
005526  104012                  AT34B:  CHAIN                   ;INTERRUPT, CHAIN
005530  000765                          BR      AT34A           ;REPEAT TEST.
005532  022626                  AT34C:  POPSP2                  ;HERE IF INTERRUPT OCCURS, POP THE
005534  000774                          BR      AT34B           ;STACK TWICE.
                                ;TEST THAT PUNCH READY DOES NOT CAUSE AN INTERRUPT WITH PROCESSOR
                                ;AT SAME PRIORITY LEVEL AS THE PUNCH INTERRUPT REQUEST LEVEL.
005536  000035                  AT35:   35                      ;TEST #
005540  005612                          AT36                    ;NEXT TEST
005542  001750                          1000.                   ;I COUNT
005544  005552                          AT35A                   ;SCOPE ENTRY
005546  104007                          STPCHV                  ;SET PUNCH INTERRUPT SERVICE
005550  005604                          AT35E                   ;TO AT35E.
005552  016767  173446  172216  AT35A:  MOV     TPLVL,PSW       ;SET PROCESSOR TO SAME PRIORITY AS PUNCH.
005560  005077  173426                  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS.
005564  052777  000100  173420          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS.
005572  000240                          NOP
005574  005077  173412          AT35B:  CLR     @TPS            ;OK IF NO INTERRUPT OCCURS.
005600  104012                          CHAIN                   ;CHAIN
005602  000763                          BR      AT35A           ;REPEAT TEST.
005604  022626                  AT35E:  POPSP2                  ;ERROR, PUNCH INTERRUPTED WITH PROCESSOR
005606  104003                          ERROR                   ;SET TO SAVE PRIORITY AS THE PUNCH.
005610  000771                          BR      AT35B
```

```
                                ;TEST THAT THE PUNCH INTERRUPTS WITH PROCESSOR AT PRIORITY ONE LEVEL LOWER
                                ;THAN THE PUNCH PRIORITY.
005012  000036                 AT36:   36                      ;TEST #
005014  005674                         AT37                    ;NEXT TEST
005016  001750                         1000.                   ;I COUNT
005020  005626                         AT36A                   ;SCOPE ENTRY
005022  104037                         STPCHV                  ;SET PUNCH INTERRUPT SERVICE
005024  005662                         AT36B                   ;TO AT36B.
005026  005077  173360         AT36A:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
005032  016767  173360  172136         MOV     TPLVL,PSW       ;SET PROCESSOR PRIORITY ONE LEVEL
005040  162767  000040  172130         SUB     #40,PSW         ;LOWER THAN PUNCH PRIORITY
005046  052777  000100  173336         BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
005054  000240                         NOP
005056  104003                         ERROR                   ;ERROR', PUNCH FAILED TO INTERRUPT.
005060  000401                         BR      AT36C
005062  022626                 AT36B:  POPSP2                  ;HERE IF INTERRUPT OCCURS, POP
005064  005077  173322         AT36C:  CLR     @TPS            ;THE STOCK TWICE, DISABLE PUNCH INTERRUPT
005070  104012                         CHAIN                   ;CHAIN
005072  000755                         BR      AT36A           ;REPEAT TEST.
                                ;TEST THAT PUNCH READY DOES NOT REINTERRUPT AFTER RTI WHEN READY
                                ;BIT HAS NOT BEEN RESET.
005074  000037                 AT37:   37                      ;TEST #
005076  005772                         AT4B                    ;NEXT TEST
005100  001750                         1000.                   ;I COUNT
005102  005724                         AT37A                   ;SCOPE ENTRY
005104  104027                 AT37A:  STPCHV                  ;SET PUNCH INTERRUPT SERVICE TO
005106  005742                         AT37C                   ;AT37C
005110  005077  173276                 CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
005114  005067  172856                 CLR     PSW             ;SET PROCESSOR PRIORITY TO 0
005120  052777  000100  173264         BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
005126  000240                         NOP
005130  104023                 AT37E1: ERROR                   ;ERROR 1, PUNCH FAILED TO INTERRUPT.
005132  005077  173254         AT37B:  CLR     @TPS            ;DISABLE PUNCH INTERRUPT.
005136  104012                         CHAIN                   ;CHAIN
005140  000761                         BR      AT37A           ;REPEAT TEST.
005142  012777  005762  173252 AT37C:  MOV     #AT37E2,@TPVTR  ;HERE IF INTERRUPT OCCURS, CHANGE
005150  012716  005156                 MOV     #AT37D,@#6      ;PUNCH VECTOR TO AT37E2 AND EXIT
005154  000002                         RTI                     ;INTERRUPT
005156  000240                 AT37D:  NOP                     ;OK IF NO REINTERRUPT OCCURS
005160  000764                         BR      AT37B
005162  022626                 AT37E2: POPSP2                  ;ERROR 2,  PUNCH REINTERRUPTED AFTER
005164  104023                         ERROR                   ;RTI WITH READY BIT LEFT ON
005166  000761                         BR      AT37B
```

```
                                ;TEST THAT THE PUNCH INTERRUPTS IMMEDIATELY UPON LOWERING
                                ;PROCESSOR PRIORITY TO 0.
005770  000040                  AT40:   40                      ;TEST #
005772  006054                          AT41                    ;NEXT TEST
005774  001750                          1000.                   ;I COUNT
005776  006004                          AT40A                   ;SCOPE ENTRY
006000  104007                          STPCHV                  ;SET PUNCH INTERRUPT
006002  006042                          AT40B                   ;SERVICE TO AT40B
006004  012767  000340  171764  AT40A:  MOV     #PRTY7,PSW      ;SET PROCESSOR PRIORITY TO 7.
006012  005077  173174                  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
006016  052777  000100  173166          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
006024  005067  171746                  CLR     PSW             ;LOWER PROCESSOR PRIORITY TO 0.
006030  012767  000340  171740          MOV     #PRTY7,PSW      ;RAISE PRIORITY TO 7.
006036  104003                          ERROR                   ;ERROR, PUNCH FAILED TO INTERRUPT
006040  000401                          BR      AT40C           ;IMMEDIATELY AFTER CP PRIORITY WAS SET TO 0.
006042  022626                  AT40B:  POPSP2                  ;HERE IF INTERRUPT OCCURS
006044  005077  173142          AT40C:  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
006050  104012                          CHAIN                   ;CHAIN
006052  000754                          BR      AT40A           ;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.
006054  000041                  AT41:   41                      ;TEST#
006056  006142                          AT42                    ;NEXT TEST
006060  000062                          50.                     ;I COUNT
006062  006074                          AT41A                   ;SCOPE ENTRY
006064  104400                          DELAY                   ;WAIT 150 MSECS
006066  000226                          150.
006070  104007                          STPCHV                  ;SET PUNCH INTERRUPT SERVICE
006072  006132                          AT41C                   ;TO AT41C
006074  012767  000001  000016  AT41A:  MOV     #WAIT,AT41B     ;MOVE WAIT INSTRUCTION TO AT41B
006102  005077  173106                  CLR     @TPB            ;LOAD PUNCH BUFFER (ENABLES PUNCH)
006106  052777  000100  173076          BIS     #BIT6,@TPS      ;ENABLE PUNCH INTERRUPTS
006114  005067  171656                  CLR     PSW             ;SET PRIORITY 0.
006120  000000                  AT41B:  OPEN                    ;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.
006122  005077  173064                  CLR     @TPS            ;DISABLE PUNCH INTERRUPTS
006126  104012                          CHAIN                   ;CHAIN
006130  000761                          BR      AT41A           ;REPEAT TEST
006132  012767  104003  177760  AT41C:  MOV     #ERROR,AT41B    ;MOVE ERROR CALL TO AT41B.
006140  000002                          RTI                     ;EXIT INTERRUPT.
```

```
                                ;TEST THAT LOADING THE PUNCH BUFFER WITH THE MAINTENANCE BIT SET
                                ;CAUSES THE READER DONE BIT TO SET AFTER APPROX, 200 MSECS
000142  000042          AT42:   42                      ;TEST #
000144  177777                  -1                      ;LAST TEST
000146  000062                  50.                     ;I COUNT
000150  000156                  AT42A                   ;SCOPE ENTRY
000152  104400                  DELAY
000154  000226                  150.
000156  052777  000004  173026  AT42A1: BIS     #BIT2,@TPS      ;SET MAINTENANCE BIT
000164  005077  173024          CLR     @TPB            ;LOAD PUNCH BUFFER
000170  104400                  DELAY                   ;WAIT 200 MSECS
000172  000310                  200.
000174  105777  173006          TSTB    @TKS            ;TEST READER DONE BIT
000200  100401                  BMI     AT42B           ;BRANCH IF READER DONE BIT SET',
000202  104003          AT42E1: ERROR                   ;ERROR, 200 MSECS AFTER PUNCH
                                ;BUFFER LOAD WITH MAINTENANCE BIT
                                ;SET THE READER DONE BIT WAS NOT SET
000204  104012          AT42B1: CHAIN                   ;CHAIN
000206  000763                  BR      AT42A           ;REPEAT TEST
```

```
                                ;PRG2-PRINTER TESTS
006210  012767  006242  173014  PRG1:   MOV     #CT0,KSTART     ;SET ADDRESS IF 1ST ROUTINE,
006216  052767  000200  173022          BIS     #BIT7,PRGID     ;BYPASS SCOPE AND ICNT,
006224  012767  177600  174770          MOV     #177600,STLMSK  ;SET STALL LIMIT
006232  004767  176046                  JSR     %7,STBF         ;SET UP BUFFER AREA,
006236  000167  173330                  JMP     SRSET           ;GO GET STARTED,
                                ;CARRIAGE RETURN TEST,
006242  000000                  CT0:    0                       ;TEST#
006244  006344                          CT1                     ;NEXT TEST ADDRESS,
006246  104000                          TYPE                    ;TYPE TITLE,
006250  011106                          CRTST
006252  012767  000110  173062          MOV     #72.,,RCNT
006260  016767  173056  173070          MOV     RCNT,CTRA       ;RCNT TO CTRA
006266  005367  173064          CT0A:   DEC     CTRA            ;DECREMENT CTRA
006272  001001                          BNE     CT0B            ;BRANCH IF NOT 0
006274  104012                          CHAIN                   ;0, CHAIN
006276  016767  173054  173054  CT0B:   MOV     CTRA,CTRB       ;SPACE COUNT TO CTRB,
006304  112700  000105          CT0C:   MOVB    #105,%0         ;CHARACTER
006310  004767  175532                  JSR     %7,LSPCH        ;SPACE,
006314  005367  173040                  DEC     CTRB            ;DECREMENT CTRB,
006320  001371                          BNE     CT0C            ;BRANCH IF NOT DONE SPACING,
006322  112700  000015                  MOVB    #15,%0
006326  004767  175514                  JSR     %7,LSPCH        ;CARRIAGE RETURN,
006332  012700  000012                  MOV     #12,%0          ;LINE FEED
006336  004767  175504                  JSR     %7,LSPCH        ;
006342  000751                          BR      CT0A
                                ;RIGHT MARGIN TEST
006344  000001                  CT1:    1                       ;TEST#
006346  006410                          CT2                     ;NEXT TEST,
006350  104000                          TYPE                    ;TYPE TITLE
006352  011137                          RMTST
006354  012767  000016  172774          MOV     #14.,,CTRA      ;SET UP FOR 33/35
006362  012767  011067  000014          MOV     #RM33B,RMB
006370  104000                  CT1A:   TYPE                    ;TYPE----!
006372  011061                          RM33A
006374  005367  172756                  DEC     CTRA            ;DONE N TIMES,
006400  001373                          BNE     CT1A            ;BRANCH IF NOT N TIMES
006402  104000                          TYPE                    ;TYPE-!-,
006404  000000                  RMB:    OPEN
006406  104012                          CHAIN                   ;CHAIN,
```

```
                                        ;CURSOR RIGHT TEST
000410  000002                  CT2:    2                       ;TEST#
000412  000562                          CT3                     ;NEXT TEST
000414  104000                          TYPE                    ;TYPE TITLE,
000416  011165                          SPTST
000420  012767  000044  172730          MOV     #36.,,CTRA      ;33/35 COUNT TO CTRA.
000426  104000                  CT2A:   TYPE                    ;TYPE SPACE,\,
000430  011103                          SPTSTC
000432  005367  172720                  DEC     CTRA            ;DONE TIMES SET IN CTRA?
000436  001373                          BNE     CT2A            ;BRANCH IF NOT DONE
000440  012767  000044  172710          MOV     #36.,,CTRA      ;SET UP CTRA COUNT FOR 33/35
000446  012767  000001  172704  CT2B:   MOV     #1,CTRB
000454  016767  172700  172700  CT2C:   MOV     CTRB,CTRC
000462  112700  000015                  MOVB    #15,X0          ;CARRIAGE RETURN.
000466  004767  175354                  JSR     X7,LSPCH
000472  004767  175350                  JSR     X7,LSPCH        ;DUMMY CYCLE,
000476  005000                  CT2D:   CLR     X0              ;NULL CHAR FOR FILLER
000500  004767  175342                  JSR     X7,LSPCH        ;TRANSMIT NULL CHAR
000504  004767  175336                  JSR     X7,LSPCH        ;TRANSMIT NULL CHAR
000510  004767  175332                  JSR     X7,LSPCH        ;TRANSMIT NULL CHAR
000514  112700  000038                  MOVB    #38,X0          ;CURSOR RIGHT
000520  004767  175322                  JSR     X7,LSPCH        ;SET IN CTRC,
000524  005367  172632                  DEC     CTRC            ;DONE SPACING,
000530  001362                          BNE     CT2D            ;BRANCH IF NOT DONE SPACING,
000532  112700  000057                  MOVB    #'/,X0          ;DONE, TYPE A "/",
000536  004767  175304                  JSR     X7,LSPCH
000542  005367  172610                  DEC     CTRA            ;DONE 36 TIMES?
000546  001001                          BNE     CT2E            ;BRANCH IF NOT DONE,
000550  104012                          CHAIN                   ;DONE, CHAIN,
000552  062767  000002  172600  CT2E:   ADD     #2,CTRB         ;MODIFY CTRB FOR NEXT TRY,
000560  000735                          BR      CT2C            ;GO DO IT AGAIN,
                                        ;LINE FEED TEST
000562  000003                  CT3:    3                       ;TEST #
000564  000642                          CT4                     ;NEXT TEST,
000566  104000                          TYPE                    ;TYPE TITLE
000570  011213                          LFTST
000572  052767  040000  172446          BIS     #BIT14,PRGID    ;ALLOW STALLS,
000600  012767  000110  172550          MOV     #72.,,CTRA      ;SET 33/35 LINE FEED COUNT,
000606  112700  000134          CT3A:   MOVB    #'\,X0          ;TYPE "\"
000612  004767  175230                  JSR     X7,LSPCH
000616  112700  000012                  MOVB    #12,X0          ;LINE FEED.
000622  004767  175220                  JSR     X7,LSPCH
000626  005367  172524                  DEC     CTRA            ;DONE N TIMES?
000632  001001                          BNE     CT3B            ;BRANCH IF NOT DONE,
000634  104012                          CHAIN                   ;DONE, CHAIN
000636  104002                  CT3B:   STALL                   ;STALL
000640  000762                          BR      CT3A            ;REPEAT
```

```
                                ;SPACE TEST
006642  000004                  CT4:    4                       ;TEST#
006644  007054                          CTS                     ;NEXT TEST,
006646  012767  000011  000074          MOV     #9,,TBCNT       ;SET TAB COUNT,
006654  104014                          CK35                    ;35?
006656  104012                          CHAIN                   ;NO,
006660  004567  000040                  JSR     %5,TPBM         ;TYPE MARKERS
006664  000007                          7
006666  104000                          TYPE
006670  011035                          TBMRK+1
006672  012767  002007  172456  CT4A:   MOV     #7,CTRA         ;LINE COUNT TO CTRA
006700  005067  000046                  CLR     SPCNT           ;0 TO SPACE COUNT,
006704  004767  000044          CT4B:   JSR     %7,TABP         ;GO SPACE-TAB,
006710  005267  000036                  INC     SPCNT           ;INCREMENT SPACE COUNT,
006714  005367  172436                  DEC     CTRA            ;DONE 7 LINES?
006720  001371                          BNE     CT4B            ;BRANCH IF NOT DONE,
006722  104012                          CHAIN                   ;DONE, CHAIN,
006724  012567  172426          TPBM:   MOV     (5)+,CTRA       ;TYPE TEST TITLE,
006730  104000                          TYPE
006732  011016                          TBTST
006734  104000                  TPBMA:  TYPE                    ;TYPE MARKERS
006736  011046                          TBMRK1
006740  005367  172412                  DEC     CTRA
006744  001373                          BNE     TPBMA
006746  000205                          RTS     %5              ;EXIT,
006750  000000                  TBCNT:  OPEN                    ;TAB COUNT
006752  000000                  SPCNT:  OPEN                    ;SPACE COUNT
006754  104000                  TABP:   TYPE                    ;CRLF,
006756  011057                          CRLF
006760  016767  177764  172372          MOV     TBCNT,CTRB      ;TAB COUNT TO CTRB
006766  016767  177760  172366  TABPA:  MOV     SPCNT,CTRC      ;SPACE COUNT TO CTRC
006774  001407                          BEQ     TABPC           ;BRANCH IF SPACE COUNT IS 0,
006776  112700  000040          TABPB:  MOVB    #40,%0          ;SPACE
007002  004767  175040                  JSR     %7,LSPCH
007006  005367  172350                  DEC     CTRC            ;DECREMENT SPACE COUNT
007012  001371                          BNE     TABPB           ;BRANCH IF NOT YET 0,
007014  112700  000011          TABPC:  MOVB    #11,%0          ;TAB
007020  004767  175022                  JSR     %7,LSPCH
007024  004767  175016                  JSR     %7,LSPCH        ;DUMMY CYCLE
007030  004767  175012                  JSR     %7,LSPCH        ;DUMMY CYCLE,
007034  112700  000057                  MOVB    #'/,%0          ;TYPE "/"
007040  004767  175002                  JSR     %7,LSPCH
007044  005367  172310                  DEC     CTRB            ;DECREMENT TAB COUNT,
007050  001346                          BNE     TABPA           ;BRANCH IF NOT DONE TABBING,
007052  000207                          RTS     %7              ;DONE, EXIT,
```

```
                                        ;TYPE LINE OF CHARACTERS ABC
007054  000005                  CT51:     5                       ;TEST #
007056  007072                            CT6                     ;NEXT TEST
007060  104000                            TYPE                    ;TYPE "CHARACTER TESTS"
007062  011236                            CHRTST
007064  104016                            TYPLN3                  ;TYPE LINE
007066  010714                            A
007070  104012                            CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS DEF
007072  000006                  CT61:     6                       ;TEST #
007074  007104                            CT7                     ;NEXT TEST
007076  104016                            TYPLN3                  ;TYPE LINE
007100  010717                            D
007102  104012                            CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS GHI
007104  000007                  CT71:     7                       ;TEST#
007106  007116                            CT10                    ;NEXT TEST,
007110  104016                            TYPLN3                  ;TYPE LINE
007112  010722                            G
007114  104012                            CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS OF JKL
007116  000010                  CT101:    10                      ;TEST #
007120  007130                            CT11                    ;NEXT TEST,
007122  104016                            TYPLN3                  ;TYPELINE
007124  010725                            J
007126  104012                            CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS MNO
007130  000011                  CT111:    11                      ;TEST #
007132  007142                            CT12                    ;NEXT TEST
007134  104016                            TYPLN3                  ;TYPE LINE
007136  010730                            M
007140  104012                            CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS PQR

007142  000012                  CT121:    12                      ;TEST #
007144  007154                            CT13                    ;NEXT TEST
007146  104016                            TYPLN3                  ;TYPE LINE
007150  010733                            P
007152  104012                            CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS STU
007154  000013                  CT131:    13                      ;TEST #
007156  007166                            CT14                    ;NEXT TEST
007160  104016                            TYPLN3
007162  010736                            S
007164  104012                            CHAIN
```

```
                                        ;TYPE LINE OF CHARACTERS VWX
007166  000014                  CT141   14                      ;TEST #
007170  007200                          CT15                    ;NEXT TEST
007172  104016                          TYPLN3                  ;TYPE LINE
007174  010741                          V
007176  104012                          CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS YZ8
007200  000015                  CT151   15                      ;TEST #
007202  007212                          CT16                    ;NEXT TEST
007204  104016                          TYPLN3                  ;TYPE LINE
007206  010744                          Y
007210  104012                          CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS 123
007212  000016                  CT161   16                      ;TEST #
007214  007224                          CT17                    ;NEXT TEST
007216  104016                          TYPLN3                  ;TYPE LINE
007220  010747                          ONE
007222  104012                          CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS 456
007224  000017                  CT171   17                      ;TEST #
007226  007236                          CT20                    ;NEXT TEST
007230  104016                          TYPLN3                  ;TYPE LINE
007232  010752                          FOUR
007234  104012                          CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS 789
007236  000020                  CT201   20                      ;TEST #
007240  007250                          CT21                    ;NEXT TEST
007242  104016                          TYPLN3                  ;TYPE LINE
007244  010755                          SEVEN
007246  104012                          CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS!"#
007250  000021                  CT211   21                      ;TEST #
007252  007262                          CT22                    ;NEXT TEST
007254  104016                          TYPLN3                  ;TYPE LINE
007256  010760                          C41
007260  104012                          CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS $%&
007262  000022                  CT221   22                      ;TEST #
007264  007274                          CT23                    ;NEXT TEST
007266  104016                          TYPLN3                  ;TYPE LINE
007270  010763                          C44
007272  104012                          CHAIN                   ;CHAIN
                                        ;TYPE LINE OF CHARACTERS '()
007274  000023                  CT231   23                      ;TEST #
007276  007306                          CT24                    ;NEXT TEST
007300  104016                          TYPLN3                  ;TYPE LINE
007302  010766                          C47
007304  104012                          CHAIN                   ;CHAIN.
```

```
                                ;TYPE LINE OF CHARACTERS @@,
007306  000024                  CT241:  24                      ;TEST #
007310  007320                          CT25                    ;NEXT TEST
007312  104016                          TYPLN3                  ;TYPE LINE
007314  010771                          C52
007316  104012                          CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS @',/
007320  000025                  CT251:  25                      ;TEST #
007322  007332                          CT26                    ;NEXT TEST
007324  104016                          TYPLN3                  ;TYPE LINE
007326  010774                          C55
007330  104012                          CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS !!<
007332  000026                  CT261:  26                      ;TEST #
007334  007344                          CT27                    ;NEXT TEST
007336  104016                          TYPLN3                  ;TYPE LINE
007340  010777                          C72
007342  104012                          CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS @>?
007344  000027                  CT271:  27                      ;TEST #
007346  007356                          CT30                    ;NEXT TEST
007350  104016                          TYPLN3                  ;TYPE LINE
007352  011002                          C75
007354  104012                          CHAIN                   ;CHAIN,
                                ;TYPE LINE OF CHARACTERS @[\
007356  000030                  CT301:  30                      ;TEST #
007360  007370                          CT31                    ;NEXT TEST
007362  104016                          TYPLN3                  ;TYPE LINE
007364  011005                          C100
007366  104012                          CHAIN                   ;CHAIN
                                ;TYPE LINE OF CHARACTERS ]^AND LEFT ARROW
007370  000031                  CT311:  31                      ;TEST #
007372  007402                          CT32                    ;NEXT TEST
007374  104016                          TYPLN3                  ;TYPE LINE
007376  011010                          C135
007400  104012                          CHAIN                   ;CHAIN
                                ;TYPE 2 LINES OF ALL CHARACTERS, FIRST LINE FULL SPEED, SECOND LINE WITH STALLS,
007402  000032                  CT321:  32                      ;TEST #
007404  007440                          CT33                    ;NEXT TEST,
007406  004767  174754                  JSR     %7,FBALL        ;FILL BUFFER WITH ALL CHARACTERS,
007412  042767  040000  171626          BIC     #BIT14,PRGID    ;CLEAR STALL BIT IN PRGID
007420  004767  174564                  JSR     %7,TYPLN        ;TYPE LINE,
007424  052767  040000  171614          BIS     #BIT14,PRGID    ;SET STALL BIT IN PRGID
007432  004767  174552                  JSR     %7,TYPLN        ;TYPE LINE,
007436  104012                          CHAIN                   ;CHAIN,
```

```
                                        ;TYPE 12 LINES OF ASR33 WORST CASE PATTERN, ALTERNATE LINES WITH STALLS.
007440  000033                  CT331:  33                      ;TEST #
007442  007522                          CT34                    ;NEXT TEST
007444  104013                          CK33                    ;33?
007446  104012                          CHAIN                   ;NO, BYPASS TEST.
007450  104000                          TYPE                    ;TYPE "WORST CASE PATTERN TEST"
007452  011262                          WCPTST
007454  004767  174746                  JSR     X7,FW336        ;PATTERN TO BUFFER.
007460  012767  000006  171670          MOV     #6,CTRA         ;SET COUNT TO 6
007466  042767  040000  171552  CT33A:  BIC     #BIT14,PRGID    ;CLEAR STALL BIT IN PRGID.
007474  004767  174510                  JSR     X7,TYPLN        ;TYPE LINE
007500  052767  040000  171540          BIS     #BIT14,PRGID    ;SET STALL BIT IN PRGID.
007506  004767  174476                  JSR     X7,TYPLN        ;TYPE LINE.
007512  005367  171640                  DEC     CTRA            ;DONE 6 TIMES?
007516  001363                          BNE     CT33A           ;BRANCH IF NOT 6 TIMES YET.
007520  104012                          CHAIN                   ;DONE, CHAIN.
                                        ;TYPE 12 LINES OF ASR33 WORST CASE PATTERN, ALTERNATE LINES WITH STALLS.
007522  000034                  CT341:  34                      ;TEST #
007524  177777                          -1                      ;LAST TEST.
007526  104014                          CK33                    ;33?
007530  104012                          CHAIN                   ;NO, BYPASS TEST.
007532  104000                          TYPE                    ;TYPE "WORST CASE PATTERN TEST"
007534  011262                          WCPTST
007536  004767  174724                  JSR     X7,FW336        ;PATTERN TO BUFFER.
007542  012767  000006  171606          MOV     #6,CTRA         ;SET COUNT TO 6.
007550  042767  040000  171470  CT34A:  BIC     #BIT14,PRGID    ;CLEAR STALL BIT  IN PRGID.
007556  004767  174426                  JSR     X7,TYPLN        ;TYPE LINE.
007562  052767  040000  171456          BIS     #BIT14,PRGID    ;SET STALL BIT IN PRGID.
007570  004767  174414                  JSR     X7,TYPLN        ;TYPE LINE
007574  005367  171556                  DEC     CTRA            ;DONE 6 TIMES?
007600  001363                          BNE     CT34A           ;BRANCH IF NOT 6 TIMES YET.
007602  104012                          CHAIN                   ;DONE, CHAIN.
```

```
                                ;PRG2-KEYBOARD TEST
007004  012767  007638  171420  PRG21:  MOV     #ET0,KSTART
007012  052767  000200  171420          BIS     #BIT7,PRGID
007020  104000                          TYPE
007022  011316                          KMSG1
007024  000167  171742                  JMP     SRSET
                                ;TEST THAT PRESSING KEY SETS DONE FLAG.
007030  000000                  ET01:   0                               ;TEST #
007032  007734                          ET1                             ;NEXT TEST.
007034  012767  000005  171514          MOV     #5,CTRA
007042  104006                  ET0A:   STRDRV
007044  007700                          ET00
007046  104000                          TYPE                            ;TYPE "PRESS A KEY WITHIN 10 SECS."
007050  011334                          KMSG2
007052  052777  000100  171326          BIS     #BIT6,@TKS              ;ENABLE KYBD INTERRUPT.
007060  005067  170112                  CLR     PSW
007064  104400                          DELAY                           ;WAIT 10 SECONDS
007066  023420                          10000.
007070  104000                          TYPE                            ;TYPE "NO KEYBOARD REQUEST."
007072  011536                          KMSG6
007074  104010                          EHALT                           ;HALT.
007076  000411                          BR      ET0CA
007700  105777  171322          ET0B1:  TSTB    @TKS                    ;TEST FOR DONE BIT ON
007704  100403                          BMI     ET0C                    ;BRANCH IF DONE BIT SET.
007706  104000                          TYPE                            ;DONE BIT NOT SET, TYPE(FALSE  KEY-
007710  011564                          KMSG7                           ;BOARD OR READER INTERRUPT.
007712  104010                          EHALT                           ;HALT
007714  012716  007722          ET0C1:  MOV     #ET0CA,@X6
007720  000002                          RTI                             ;EXIT INTERRUPT.
007722  104011                  ET0CA:  SRESET
007724  005367  171426                  DEC     CTRA                    ;DONE 5 TIMES?
007730  001344                          BNE     ET0A                    ;BRANCH IF NOT DONE.
007732  104012                          CHAIN                           ;CHAIN
                                ;ECHO TEST. KEYED CHARACTER IS TYPED. RUBOUT ENDS ROUTINE.
007734  000001                  ET1:    1                               ;TEST #
007736  010014                          ET2                             ;NEXT TEST.
007740  104000                          TYPE                            ;TYPE TITLE AND INSTRUCTIONS.
007742  011374                          KMSG3
007744  105777  171236          ET1A:   TSTB    @TKS                    ;WAIT FOR DONE FLAG
007750  100375                          BPL     .-4
007752  117767  171232  171364          MOVB    @TKB,CRBUF              ;MOVE KYBD CHAR TO CRBUF.
007760  116777  171360  171226          MOVB    CRBUF,@TPB              ;ECHO CHAR READ.
007766  105777  171220                  TSTB    @TPS                    ;WAIT FOR PRINTER DONE.
007772  100375                          BPL     .-4
007774  042767  000200  171342          BIC     #BIT7,CRBUF             ;CLEAR BIT 7 FROM CRBUF.
010002  122767  000177  171334          CMPB    #177,CRBUF              ;COMPARE CRBUF TO RUBOUT (177)
010010  001355                          BNE     ET1A                    ;BRANCH IF NOT RUBOUT (177)
010012  104012                          CHAIN                           ;CHAIN
```

```
                                ;OCTAL EQUIVALENT TEST, THE OCTAL EQUIVALENT OF ANY CHARACTER KEYED
                                ;;S PRINTED, RUBOUT ENDS ROUTINE.
010014  000002                 ET2:    2                               ;TEST #
010016  177777                         -1                              ;LAST TEST
010020  104001                         TYPES                           ;TYPE TITLE AND INSTRUCTIONS.
010022  011476                         KMSG4
010024  011407                         KMSG3A
010026  177777                         -1
010030  005067  171310                 CLR     CRBUF
010034  105777  171146         ET2A:   TSTB    @TKS                    ;WAIT FOR DONE FLAG.
010040  100375                         BPL     .-4
010042  117767  171142  171274         MOVB    @TKB,CRBUF              ;CHARACTER TO CRBUF
010050  004567  173622                 JSR     %5,ACNV4                ;CONVERT CHAR IN CRBUF TO
010054  001344                         CRBUF                           ;PRINTABLE OCTAL
010056  011530                         OCTEQV
010060  104000                         TYPE                            ;TYPE OCTAL EQUIVALENT
010062  011526                         KMSG5
010064  042767  000200  171252         BIC     #BIT7,CRBUF             ;CLEAR BIT 7 FROM CRBUF
010072  022767  000177  171244         CMP     #177,CRBUF              ;TEST FOR RUBOUT CHARACTER.
010100  001355                         BNE     ET2A                    ;BRANCH IF NOT RUBOUT (177).
010102  104012                         CHAIN                           ;CHAIN.
```

```
                                ;PRG3-PRINTER EXERCISER, KEYBOARD CONTROLLED,
                                ;TYPES LINES WITH ANY 5 CHARACTERS, STALLS OR FULL SPEED.
010104  004767  174174         PRG3:  JSR     %7,STBF          ;SET UP BUFFER,
010110  104000                        TYPE                    ;TYPE TITLE
010112  011612                        P7MG1
010114  052767  040000  171124 HTA:   BIS     #BIT14,PRGID     ;SET STALL BIT IN PRGID,
010122  012767  177600  173072        MOV     #177600,STLMSK   ;SET STALL MASK
010130  012703  011673                MOV     #BLOCK1,%3
010134  104000                        TYPE                    ;TYPE "TYPE IN DATA",
010136  011640                        P7MG2
010140  012767  000006  171210        MOV     #6,CTRA          ;CHAR COUNT TO CTRA,
010146  004767  000060        HTB:    JSR     %7,GKBCR         ;GET AND STORE KYBD CHARACTER,
010152  005367  171200                DEC     CTRA             ;GOT 6 CHARACTERS?
010156  001373                        BNE     HTB              ;BRANCH IF NOT 6 CHARS YET,
010160  042767  000200  171156        BIC     #BIT7,CRBUF
010166  122767  000177  171150        CMPB    #177,CRBUF       ;CHECK 6TH CHAR FOR RUBOUT,
010174  001003                        BNE     HTC              ;BRANCH IF NOT A RUBOUT,
010176  042767  040000  171042        BIC     #BIT14,PRGID     ;RUBOUT, CLEAR STALL BIT IN PRGID,
010204  004567  173600        HTC:    JSR     %5,BMOVE         ;FILL 92 CHAR LINE,
010210  011673                        BLOCK1
010212  011700                        BLOCK1+5
010214  000103                        67,
010216  004767  173766        HTD:    JSR     %7,TYPLN         ;TYPE LINE,
010222  005767  167342                TST     SR               ;CHANGE DATA? (SR15=1),
010226  100732                        BMI     HTA              ;YES, GO CHANGE DATA
010230  000772                        BR      HTD              ;NO CONTINUE WITH SAME DATA,
010232  105777  176750        GKBCR:  TSTB    @TKS             ;WAIT FOR DONE FLAG,
010236  100375                        BPL     .-4
010240  117767  176744  171076        MOVB    @TKB,CRBUF       ;CHARACTER TO CRBUF,
010246  116723  171072                MOVB    CRBUF,(3)+       ;CHARACTER TO LINE BUFFER,
010252  116700  171066                MOVB    CRBUF,%0
010256  004767  173564                JSR     %7,LSPCH         ;ECHO CHARACTER,
010262  000207                        RTS     %7
```

```
                                        ;
                                        ;
                                        ;PRG4-PUNCH CLOCK ADJUSTMENT ROUTINE,
                                        ;OUTPUTS CHARACTER SET IN LEFT HALF OF SR, AND
                                        ;STALLS FOR NUMBER OF MILLISECONDS SET IN RIGHT HALF OF SR,
010264  104035                          PRG4:   CHALT                   ;HALT TO SET SR,
010266  004767  000036                  1$:     JSR     %7,C1112        ;GO OUTPUT CHARACTER SET IN LEFT
010272  000775                                  BR      1$              ;HALF OF SR AND STALL PER SR RIGHT,
                                        ;
                                        ;
                                        ;PRG5-READER CLOCK ADJUSTMENT ROUTINE,
                                        ;PERFORMS SAME FUNCTION AS PRG11, AND IN ADDITION,
                                        ;USING THE PUNCH MAINTENANCE BIT, SHIFTS OUTPUT OF PUNCH
                                        ;SHIFT REGISTER ONTO THE READER BUFFER, THE CONTENTS OF THE
                                        ;READER BUFFER ARE THEN "FIXED" ON THE CONSOLE DATA LIGHTS
                                        ;BY ISSUING A RESET WITH CONTENTS OF READER BUFFER LOADED IN R0,
010274  104005                          PRG5:   CHALT                   ;HALT TO SET SR,
010276  004767  000020                  1$:     JSR     %7,C1112M       ;GO OUTPUT CHARACTER FROM SR LEFT AND
010302  217700  170702                          MOV     @TKB,%0         ;STALL PER SR RIGHT, (TKB) TO R0,
010306  000005                                  RESET                   ;"FIX" (TKB) IN DATA LIGHTS,
010310  000005                                  RESET
010312  000005                                  RESET
010314  000005                                  RESET
010316  000005                                  RESET
010320  000766                                  BR      1$              ;REPEAT,
                                        ;
010322  252777  000004  170662          C1112M: BIS     #4,@TPS         ;SET MAINTENANCE MODE (PUNCH),
010330  116767  167234  000022          C1112:  MOVB    SR,XTY          ;STALL COUNT TO XTY,
010336  005767  000016                          TST     XTY             ;DISREGARD 0 DELAY,
010342  001002                                  BNE     C1112A
010344  005267  000012                          INC     XTY
010350  116777  167215  170636          C1112A: MOVB    SR+1,@TPB       ;LOAD PUNCH BUFFER,
010356  104400                                  DELAY                   ;DELAY (APPROXIMATELY) THE NUMBER OF
010360  000000                          XTY:    OPEN                    ;MSECS, SPECIFIED AT SR RIGHT
010362  000207                                  RTS     %7              ;EXIT
```

```
                                        ;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,
018364  104005                  PRG6:   CHALT                   ;HALT TO SET SR,
018366  052777  000004  170616  KTA:    BIS     #4,@TPS         ;SET MAINTENANCE MODE,
018374  105777  170612          KTB:    TSTB    @TPS            ;WAIT FOR READY,
018400  100375                          BPL     .-4
018402  116767  167163  170735          MOVB    SR+1,CRBUF+1    ;S/B CHAR TO CRBUF+1,
018410  116777  170731  170576          MOVB    CRBUF+1,@TPB    ;OUTPUT CHARACTER,
018416  105777  170564                  TSTB    @TKS            ;WAIT FOR READER DONE FLAG,
018422  100375                          BPL     .-4
018424  117767  170560  170712          MOVB    @TKB,CRBUF      ;CHAR READ TO CRBUF,
018432  104004                          DATCHK                  ;GO CHECK AGAINST S/B CHAR,
018434  000754                          BR      KTA             ;REPEAT,

                                        ;PRG7-MAINTENANCE MODE SPECIAL BINARY COUNT PATTERN DATA TEST,
                                        ;PERFORMS SAME OPERATION AS PRG13, EXCEPT THAT SPECIAL BINARY COUNT
                                        ;PATTERN IS USED,
018436  004767  173034          PRG7:   JSR     %7,INBIN        ;INITIALIZE BINARY COUNT
018442  012767  177600  172552          MOV     #177600,STLMSK  ;SET STALL LIMIT
018450  052767  040000  170570          BIS     #BIT14,PRGID    ;ALLOW STALLS
018456  052777  000004  170526  LTA:    BIS     #4,@TPS         ;SET MAINTENANCE MODE,
018464  032767  000400  167076          BIT     #BIT8,SR        ;CHECK STALL SWITCH
018472  001001                          BNE     LTB             ;BRANCH IF NO STALL WANTED
018474  104002                          STALL                   ;STALL
018476  105777  170510          LTB:    TSTB    @TPS            ;WAIT FOR READY,
018502  100375                          BPL     .-4
018504  004767  173072                  JSR     %7,GTBINP       ;GET BIN CHARACTER,
018510  110167  170631                  MOVB    %1,CRBUF+1      ;MOVE TO S/B CHAR,
018514  110177  170474                  MOVB    %1,@TPB         ;OUTPUT BIN CHARACTER,
018520  105777  170462                  TSTB    @TKS            ;WAIT FOR READER DONE,
018524  100375                          BPL     .-4
018526  117767  170456  170610          MOVB    @TKB,CRBUF      ;CHAR IN READ BUFFER TO CRBUF,
018534  104004                          DATCHK                  ;GO CHECK AGAINST S/B CHAR,
018536  000747                          BR      LTA             ;CONTINUE,

                                        ;PRG10 ROLE UP TEST
                                        ;THE FUNCTION OF THIS TEST IS TO TEST THE ROLL-UP CAPABILITY
                                        ;OF THE VT36
                                        ;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

018540  012767  177736  000126  PRG10:  MOV     #-42,TCHAR      ;INIT TEMP CHAR
018546  016767  000122  000122  RENIT:  MOV     TCHAR,CHAR      ;COMPLEMENT OF "!"
018554  012767  177670  000110  PRG10C: MOV     #-72,,CNT       ;72 CHAR/LINE
018562  005167  000110          PRG10D: COM     CHAR            ;
```

```
018566  016700  000104          PRG10A: MOV     CHAR,R0         ;LOAD "!"
018572  004767  173258                  JSR     R7,LSPCH        ;PUNCH "!"
018576  005167  000074                  COM     CHAR            ;COMPLEMENT TO "."
019002  016700  000070                  MOV     CHAR,R0         ;LOAD "."
019006  004767  173234                  JSR     R7,LSPCH        ;PUNCH "."
019012  005167  000060                  COM     CHAR            ; "!"
019016  062767  000002  000046          ADD     #2,CNT          ;END OF LINE?
019024  001360                          BNE     PRG10A          ;NO
019026  012700  000015                  MOV     #15,R0          ;CR
019032  004767  173210                  JSR     R7,LSPCH
019036  012700  000012                  MOV     #12,R0          ;LF
019042  004767  173200                  JSR     R7,LSPCH
019046  005767  166716                  TST     SR              ;NEXT CHAR
019052  100340                          BPL     PRG10C          ;NO
019054  005367  000014                  DEC     TCHAR           ;YES CHANGE TCHAR
019060  022767  177677  000006          CMP     #177677,TCHAR   ;CHAR STRING COMPLETE
019066  001724                          BEQ     PRG10
019070  000726                          BR      RENIT
019072  000000          CNT:    0
019074  177736          TCHAR:  -42
019076  000041          CHAR:   41
```

```
018700     047                A33WP61 ,BYTE   047,137,127,057,127,137
018701     137
018702     127
018703     057
018704     127
018705     137
018706     047                A33WP61 ,BYTE   047,133,077,103,077,133
018707     133
018710     077
018711     103
018712     077
018713     133
018714     101                A1      ,BYTE   101,102,103
018715     102
018716     103
018717     104                D1      ,BYTE   104,105,106
018720     105
018721     106
018722     107                G1      ,BYTE   107,110,111
018723     110
018724     111
018725     112                J1      ,BYTE   112,113,114
018726     113
018727     114
018730     115                M1      ,BYTE   115,116,117
018731     116
018732     117
018733     120                P1      ,BYTE   120,121,122
018734     121
018735     122
018736     123                S1      ,BYTE   123,124,125
018737     124
018740     125
018741     126                V1      ,BYTE   126,127,130
018742     127
018743     130
018744     131                Y1      ,BYTE   131,132,060
018745     132
018746     060
018747     061                ONE1    ,BYTE   061,062,063
018750     062
018751     063
018752     064                FOUR1   ,BYTE   064,065,066
018753     065
018754     066
018755     067                SEVEN1  ,BYTE   067,070,071
018756     070
018757     071
018760     041                C411    ,BYTE   041,042,043
018761     042
018762     043
018763     044                C441    ,BYTE   044,045,046
018764     045
018765     046
```

```
011766      047                 C47:    .BYTE   047,050,051
011767      050
011770      051
011771      052                 C52:    .BYTE   052,053,054
011772      053
011773      054
011774      055                 C55:    .BYTE   055,056,057
011775      056
011776      057
011777      072                 C72:    .BYTE   072,073,074
012000      073
012001      074
012002      075                 C75:    .BYTE   075,076,077
012003      076
012004      077
012005      100                 C100:   .BYTE   100,133,134
012006      133
012007      134
012010      135                 C135:   .BYTE   135,136,137
012011      136
012012      137
012013      377                 C377:   .BYTE   377,200,377
012014      200
012015      377
012016      045                 TBTST:  .ASCII  '%%SPACE TES%%%'
012017      043
012020      123
012021      120
012022      121
012023      103
012024      105
012025      040
012026      124
012027      105
012030      123
012031      124
012032      045
012033      043

012034      040                 TBMRK:  .ASCII  '         /0'
012035      040
012036      040
012037      040
012040      040
012041      040
012042      040
012043      040
012044      057
012045      100

012046      040                 TBMRK1: .ASCII  '         /0'
012047      040
012050      040
012051      040
012052      040
```

```
011053      040
011054      040
011055      057
011056      100

011057      045                 CRLF:   .ASCII  '%0'
011060      100

011061      055                 RM33A:  .ASCII  '-----10'
011062      055
011063      055
011064      055
011065      111
011066      100

011067      055                 RM33B:  .ASCII  '-1-0'
011070      111
011071      055
011072      120

011073      055                 RM37A:  .ASCII  '-----1-10'
011074      055
011075      055
011076      055
011077      111
011100      055
011101      111
011102      100

011103      134                 SPTSTC: .ASCII  '\ 0'
011104      040
011105      100

011106      045                 CRTST:  .ASCII  '%%CARRIAGE RETURN TEST%%0'
011107      043
011110      103
011111      101
011112      122
011113      122
011114      111
011115      101
011116      107
011117      105
011120      040
011121      122
011122      105
011123      124
011124      125
011125      122
011126      116
011127      040
011130      124
011131      125
011132      123
011133      124
```

```
011134      045
011135      043
011136      100

011137      045                 RMTST:  .ASCII  '%@RIGHT MARGIN TEST%00'
011140      043
011141      122
011142      111
011143      107
011144      110
011145      124
011146      040
011147      115
011150      101
011151      122
011152      107
011153      111
011154      110
011155      040
011156      124
011157      105
011160      123
011161      124
011162      045
011163      043
011164      100

011165      045                 SPTST:  .ASCII  '%@CURSOR RIGHT TEST%00'
011166      043
011167      103
011170      125
011171      122
011172      123
011173      117
011174      122
011175      040
011176      122
011177      111
011200      107
011201      110
011202      124
011203      040
011204      124
011205      125
011206      123
011207      124
011210      045
011211      043
011212      100

011213      045                 LFTST:  .ASCII  '%@LINE FEED TEST%00'
011214      043
011215      114
011216      111
011217      116
```

```
011220      105
011221      040
011222      106
011223      105
011224      105
011225      104
011226      040
011227      124
011230      105
011231      123
011232      124
011233      045
011234      043
011235      100

011236      045                 CHRTST: .ASCII  '%%CHARACTER TEST%%00'
011237      043
011240      103
011241      110
011242      101
011243      122
011244      101
011245      103
011246      124
011247      105
011250      122
011251      040
011252      124
011253      105
011254      123
011255      124
011256      123
011257      045
011260      043
011261      100

011262      045                 WCPTST: .ASCII  '%%WORST CASE PATTERN TEST%%00'
011263      043
011264      127
011265      117
011266      122
011267      123
011270      124
011271      040
011272      103
011273      101
011274      123
011275      105
011276      040
011277      120
011300      101
011301      124
011302      124
011303      105
011304      122
```

```
011305      116
011306      040
011307      124
011310      105
011311      123
011312      124
011313      045
011314      043
011315      100

011316      045                 KMSG11  .ASCII  '%@KYBD TEST%@@'
011317      043
011320      113
011321      131
011322      102
011323      124
011324      040
011325      124
011326      105
011327      123
011330      124
011331      045
011332      043
011333      100

011334      045                 KMSG21  .ASCII  '%PRESS A KEY WITHIN 10 SECONDS,@'
011335      120
011336      122
011337      105
011340      123
011341      123
011342      040
011343      101
011344      040
011345      113
011346      105
011347      131
011350      040
011351      127
011352      111
011353      124
011354      110
011355      111
011356      116
011357      040
011360      061
011361      060
011362      040
011363      123
011364      105
011365      103
011366      117
011367      116
011370      104
011371      123
```

```
011372      056
011373      100

011374      045                 KMSG31  ,ASCII  /%#ECHO TEST/
011375      043
011376      105
011377      103
011400      110
011401      117
011402      040
011403      124
011404      105
011405      123
011406      124

011407      045                 KMSG3A1 ,ASCII  /%CHARACTER KEYED WILL BE TYPED./
011410      103
011411      110
011412      101
011413      122
011414      101
011415      103
011416      124
011417      105
011420      122
011421      040
011422      113
011423      105
011424      131
011425      105
011426      104
011427      040
011430      127
011431      111
011432      114
011433      114
011434      040
011435      102
011436      105
011437      040
011440      124
011441      131
011442      120
011443      105
011444      104
011445      056

011446      045                 ,ASCII  /%RUBOUT ENDS ROUTINE,%#0/
011447      122
011450      125
011451      102
011452      117
011453      125
011454      124
011455      040
```

```
011456      105
011457      116
011460      104
011461      123
011462      040
011463      122
011464      117
011465      125
011466      124
011467      111
011470      116
011471      105
011472      056
011473      045
011474      043
011475      100

011476      045                 KMSG4:  .ASCII  /%#OCTAL EQUIVALENT TEST@/
011477      043
011500      117
011501      103
011502      124
011503      101
011504      114
011505      040
011506      105
011507      121
011510      125
011511      111
011512      126
011513      101
011514      114
011515      105
011516      116
011517      124
011520      040
011521      124
011522      105
011523      123
011524      124
011525      100

011526      045                 KMSG5:  .ASCII  /% /
011527      040

011530      040                 OCTEQV: .ASCII  /    %@/
011531      040
011532      040
011533      040
011534      045
011535      100

011536      045                 KMSG6:  .ASCII  /%NO KEYBOARD REQUEST,@/
011537      116
011540      117
```

```
011941      040
011942      113
011943      105
011944      131
011945      102
011946      117
011947      101
011950      122
011951      104
011952      040
011953      122
011954      105
011955      121
011956      125
011957      105
011960      123
011961      124
011962      056
011963      100

011964      045                 KMSG71  ,ASCII  'XFALSE KYBD INTERRUPT@'
011965      106
011966      101
011967      114
011970      123
011971      105
011972      040
011973      113
011974      131
011975      102
011976      104
011977      040
012000      111
012001      110
012002      124
012003      105
012004      122
012005      122
012006      125
012007      120
012010      124
012011      100

012012      045             P7MG11  ,ASCII  'X@DISPLAY EXERCISERX@@'
012013      043
012014      104
012015      111
012016      123
012017      120
012020      114
012021      101
012022      131
012023      040
012024      105
012025      130
```

```
011026      105
011027      122
011030      103
011031      111
011032      123
011033      105
011034      122
011035      045
011036      043
011037      100

011040      045                 P7MG21  ,ASCII  /X-TYPE IN DATA IS/
011041      043
011042      124
011043      131
011044      120
011045      105
011046      040
011047      111
011050      116
011051      040
011052      104
011053      101
011054      124
011055      101
011056      040
011057      072
011060      100

011061      125                 BKSUI   ,ASCII  /U =/
011062      040
011063      100

011064      040                 DECVAL: ,ASCII  /     /
011065      040
011066      040
011067      040
011070      040

            000001              DEND:   ,END
```

```
A       010714          AT31    005330          AT4E2   004766          C52     010771
A1ST    003724          AT31A   005340          AT5     004774          C55     010774
A33WP6  010700          AT31B   005362          AT5A    005012          C72     010777
A35WP6  010720          AT31E   005360          AT5B    005034          C75     011002
ACNV    003734          AT32    005366          AT5E    005032          CC      177776
ACNV4   003676          AT32A   005376          BCHECK  003250          CHAIN   104012
ACNV6   003650          AT32B   005424          BDCNV   004070          CHAINN  001672
ACNV8   003670          AT32E   005422          BDCNVA  004110          CHALT   104005
ACNVC   003716          AT33    005430          BELL    000007          CHAR    010676
ACNVM   003750          AT33A   005440          BIT0    002000          CHK33   002256
ACNVX   003732          AT33B   005464          BIT1    000002          CHK33B  002270
ADTENP  004176          AT33E   005462          BIT10   002000          CHK35   002272
ARDA    002446          AT34    005470          BIT11   004000          CHKASR  104022
ARDB    002472          AT34A   005504          BIT12   010000          CHLT    001366
AREAD   002434          AT34B   005526          BIT13   020000          CHNA    001726
AT0     004540          AT34C   005532          BIT14   040000          CHNAA   001754
AT0A    004556          AT34E   005524          BIT15   100000          CHNB    001762
AT0B    004562          AT35    005536          BIT2    000004          CHR1    001346
AT0E    004566          AT35A   005552          BIT3    000010          CHR2    001350
AT1     004572          AT35B   005574          BIT4    000020          CHR3    001352
AT1A    004610          AT35E   005604          BIT5    000040          CHRTST  011236
AT1B    004614          AT36    005612          BIT6    000100          CK33    104013
AT1E    004620          AT36A   005626          BIT7    000200          CK35    104014
AT2     004624          AT36B   005662          BIT8    000400          CKASR   002310
AT24    005040          AT36C   005664          BIT9    001000          CLEAN   002122
AT24A   005056          AT37    005674          BKSU    011661          CNT     010672
AT24B   005100          AT37A   005724          BLK2    012016          CNVCTR  004170
AT24C   005120          AT37B   005732          BLKBB   012014          CPRDY   004032
AT24E1  005074          AT37C   005742          BLKCC   012126          CPRDYA  004042
AT24E2  005116          AT37D   005756          BLOCK1  011693          CRBUF   001344
AT25    005124          AT37E1  005730          BLOCK2  012005          CRLF    011057
AT25A   005142          AT37E2  005762          BLOCKA  011691          CRTA    001554
AT25B   005164          AT3A    004674          BLOCKB  012003          CRTST   011106
AT25E   005162          AT3B    004700          BLOCKC  012115          CT0     006242
AT26    005170          AT3E    004704          BMOVA   004020          CT0A    006266
AT26A   005200          AT4     004710          BMOVE   004010          CT0B    006276
AT26B   005222          AT40    005770          BRCTR   002742          CT0C    006304
AT26C   005242          AT40A   006004          BREAD   002644          CT1     006344
AT26E1  005216          AT40B   006042          BREADA  002722          CT10    007116
AT26E2  005240          AT40C   006044          BREADB  002712          CT11    007130
AT27    005246          AT41    006054          BREADC  002732          CT12    007142
AT27A   005256          AT41A   006074          BSYNC   003310          CT13    007154
AT27B   005300          AT41B   006120          BYPMAN  000400          CT14    007166
AT27E   005276          AT41C   006132          C100    011005          CT15    007200
AT2A    004642          AT42    006142          C1112   012330          CT16    007212
AT2B    004646          AT42A   006156          C1112A  012350          CT17    007224
AT2E    004652          AT42B   006204          C1112M  012322          CT1A    006370
AT3     004656          AT42E   006202          C135    011010          CT2     006410
AT30    005304          AT4A    004726          C377    011013          CT20    007236
AT30A   005314          AT4B    004750          C41     010700          CT21    007250
AT30B   005324          AT4C    004770          C44     010703          CT22    007262
AT30E   005322          AT4E1   004744          C47     010706          CT23    007274
```

```
CT24    007306          ERROR   104003          KSTART  001232          P?1     003524
CT25    007320          ET0     007630          KTA     010366          P?1P    003532
CT26    007332          ET0A    007642          KTB     010394          RCMSK   003244
CT27    007344          ET0B    007700          LFTST   011213          RCNT    001342
CT2A    006426          ET0C    007714          LPRGSW  002000          RENIT   010546
CT2B    006446          ET0CA   007722          LSPCW   004046          RIND    003520
CT2C    006454          ET1     007734          LTA     010456          RM33A   011061
CT2D    006476          ET1A    007744          LTB     010476          RM33B   011067
CT2L    006552          ET2     010014          M       010730          RM37A   011073
CT3     006562          ET2A    010034          MACHER  000004          RMB     006404
CT30    007356          FBALL   004366          MANUAL  100000          RMTST   011137
CT31    007370          FBF3    004322          NITHSW  004000          RNCNT   003246
CT32    007402          FBF3A   004332          NOP     000240          RNGEN   002572
CT33    007440          FBF3B   004340          NPRTSW  020000          RP1     002640
CT33A   007466          FORWD   002010          NTRCSW  010000          RP2     002642
CT34    007522          FORWDA  002050          NXTST   001240          RSTPC   002430
CT34A   007550          FORWDB  002056          OCTEQV  011530          RSTPSW  002432
CT3A    006606          FOUR    010752          ONE     010747          RSTREG  104021
CT3B    006636          FW336   004426          OPEN    000000          RSTRG   002374
CT4     006642          FW356   004466          P       010733          R?NNO   001236
CT4A    006672          G       010722          P?MG1   011612          S       010736
CT4B    006704          GETRDY  001574          P?MG2   011640          SAVREG  104020
CT5     007054          GKBCR   010232          PIND    003526          SAVRG   002334
CT6     007072          GOTST   002074          POPSP   005726          SCOPSW  040000
CT7     007104          GOTSTA  002114          POPSP2  022626          SCOPTR  001244
CTRA    001356          GRCNT   003224          PRG0    004526          SEVEN   010755
CTRB    001360          GTBIN   003534          PRG1    006210          SHALT   001472
CTRC    001362          GTBINP  003602          PRG10   010540          SHLTA   001506
CTRD    001364          GTRDYA  001606          PRG10A  010566          SPBOT   001204
CURIST  001234          GTRDYB  001612          PRG10C  010554          SPCNT   006752
D       010717          GTRDYC  001630          PRG10D  010562          SPTST   011165
DATCHK  104004          GTRDYD  001656          PRG2    007604          SPTSTC  011103
DATHLT  104017          HLTSW   100000          PRG3    010104          SR      177570
DECVAL  011664          HTA     010114          PRG4    010264          SRESET  104011
DELAY   104400          HTB     010146          PRG5    010274          SRSET   001572
DEND    011671          HTC     010204          PRG6    010364          SRSETT  002554
DIGIT   004172          HTD     010216          PRG7    010436          SRTSW   001000
DLCNT   003162          ICTR    001242          PRGEND  002012          S?AL    003164
DLY     003120          INBIN   003476          PRGID   001246          S?ALA   003216
DLYA    003140          INCPRG  001550          PRGNUM  001230          S?ALAA  003176
DLYB    003144          INCRTN  001666          PRGTAB  001250          S?ALB   003220
DTCHK   001412          ITA     010266          PRTY0   000000          S?ALL   104002
DTCHKA  001430          J       010725          PRTY1   000040          S?ART   001510
DTHLT   001422          JTA     010276          PRTY2   000100          S?BF    004304
EHALT   104010          KMSG1   011316          PRTY3   000140          S?LMSK  003222
EHLT    001400          KMSG2   011334          PRTY4   000200          S?LSPV  002524
EMTA    002172          KMSG3   011374          PRTY5   000240          S?LSRV  002474
EMTINT  002146          KMSG3A  011407          PRTY6   000300          S?PCHV  104007
EMTTAB  001272          KMSG4   011476          PRTY7   000340          S?PPA   002542
ERCTR   001354          KMSG5   011526          PSW     177776          S?PRA   002512
ERR     001432          KMSG6   011536          PT0     003522          S?RDRV  104006
ERRA    001452          KMSG7   011564          PT0P    003530          SUBTEN  004130
```

```
SUBTNA  004134          TYPSA   003112
SUBTNB  004150          TYPSB   003114
SVRPC   002370          V       010741
SVRPSW  002372          WCPTST  011262
SYCTRA  003472          XTY     010360
SYCTRB  003474          Y       010744
SYNCA   003370
SYNCB   003404
SYNCC   003436
SYNCD   003442
SYNCE   003464
TABP    006754
TABPA   006766
TABPB   006776
TABPC   007014
TBCNT   006750
TBMHK   011034
TBMHK1  011046
TBTST   011016
TCHAR   010674
TCTH    004242
TENPWR  004174
TKB     001210
TKLVL   001220
TKS     001206
TKVTR   001216
TPB     001214
TPBM    006724
TPBMA   006734
TPLJA   004266
TPLVL   001224
TPS     001212
TPVTR   001222
TRPA    002236
TRPINT  002212
TRPTAB  001340
TTYTYP  001226
TYP     002744
TYPA    002754
TYPC    002772
TYPJ    003020
TYPJAT  003064
TYPE    104020
TYPES   104001
TYPF    003036
TYPL    003050
TYPLJ   004244
TYPLA   004216
TYPLB   004222
TYPLN   004210
TYPLNJ  104016
TYPS    003066
```

```
ERRORS DETECTED: 0
RUN-TIME:  24 SECONDS
5K CORE USED
```