```
                .REM !
```

IDENTIFICATION

PRODUCT CODE:	AC-8472C-MC
PRODUCT NAME:	CZDHHCO DH11 AUTO-ECHO TEST
DATE:	JUNE 1985
MAINTAINER:	NAC SOFTWARE ENGINEERING
AUTHOR:	G. BAISLEY

1.0 ABSTRACT

VERIFIES THAT THE AUTO ECHO FEATURE OF THE DH11 WORKS.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 FAMILY STANDARD COMPUTER WITH 4KW OF MEMORY
ASR-33 TELETYPE OR EQUIVALENT
DH11 ASYNCHRONOUS MULTIPLEXER
DM11 MAINTENANCE CARD INSTALLED

2.2 STORAGE

THE PROGRAM LOADS INTO 4KW OF MEMORY

3.0 LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES IS TO BE USED

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)

ALL CONSOLE SWITCHES DOWN

4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES AFTER PROGRAM RESTART

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER PROGRAM RESTART

SW01=1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200
THE RESTART ADDRESS FOR ALL TESTS IS 000200
THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY

4.3.1.2 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

4.3.1.5 THE PROGRAM WILL TYPE "DH11 AUTO-ECHO TEST" AND WILL THEN TYPE "VECTOR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD.

4.3.1.6 TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR FOR THE DH11 TO BE TESTED FOLLOWED BY <CARRIAGE RETURN>

NOTE

WORDS IN ANGLE BRACKETS, I.E. <CARRIAGE RETURN> MEAN THAT THE TELETYPE KEY WITH THE NAMED FUNCTION SHOULD BE STRUCK

IF AN INCORRECT ADDRESS IS ENTERED, THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE SECOND MESSAGE OF 4.3.1.5

4.3.1.7 THE PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.1.8 TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER OF THE DH11 TO BE TESTED FOLLOWED BY <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED, THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE OF 4.3.1.7

4.3.1.9 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT TO START TESTING, AND THEN TESTING WILL BEGIN

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 PERFORM 4.3.1.2 TO 4.3.1.5

4.3.2.2 THE PROGRAM WILL TYPE "DH11 AUTO-ECHO TEST" AND WILL THEN CONTINUE AS DESCRIBED IN 4.3.1.9

4.3.3 PRDGRAM RESTART WITH SW00=1

4.3.3.1 LOAD ADDRESS 000200

4.3.3.2 SET SW01=1

4.3.3.3 PRESS START

4.3.3.4 THE PROGRAM WILL PERFORM AS DESCRIBED IN 4.3.1.5 TO 4.3.1.9

4.3.4 PROGRAM RESTART WITH SW01=1

4.3.4.1 LOAD ADDRESS 000200

4.3.4.2 SET SW01=1

4.3.4.3 PRESS START

4.3.4.4 THE PROGRAM WILL TYPE "DH11 AUTO-ECHO TEST" AND WILL THEN TYPE "TEST PC-" AND WILL WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO BE STARTED FOLLOWED BY <CARRIAGE RETURN>

4.3.4.6 THE PROGRAM WILL TYPE R TO INDICATE THAT IT HAS STARTED AND WILL START TESTING AT THE SELECTED TEST.

NOTE

CARE MUST BE TAKEN WHEN THIS FEATURE IS USED, SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS THAT IS IN THE MIDDLE OF A TEST

NOTE

IF IT IS DESIRED TO LOOP ON THE TEST
THAT IS SELECTED SET SW14=1 BEFORE
ENTERING THE TEST ADDRESS

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SW15=1, HALT ON ERROR
SW14=1, LOOP ON CURRENT TEST
SW13=1, SUPPRESS ERROR TYPEOUT
SW11=1, INHIBIT ITERATIONS
SW10=1, ESCAPE TO NEXT TEST ON ERROR
SW09=1, FREEZE VARIABLE PARAMETER IN CURRENT TEST
SW01=1, START PROGRAM AT SELECTED TEST
SW00=1, CHANGE PARAMETERS AT PROGRAM RESTART

5.2 SUBROUTINE ABSTRACTS

5.2.1 TRAPCATCHER (LOCATIONS 000000-000776)

THIS ROUTINE IS USED TO INTERCEPT UNEXPECTED INTERRUPTS AND TRAPS.
THE AREA FROM 000000-000776 IS LOADED WITH THE FOLLOWING SEQUENCE

```
2
0
4
0
...
772
0
776
0
```

IF AN UNEXPECTED INTERRUPT OR TRAP OCCURS, THE PROGRAM WILL HALT WITH
THE PC 2 GREATER THAN THE ADDRESS TO WHICH THE PROGRAM TRAPPED. THE
PROCESSOR STACK MAY BE EXAMINED TO DETERMINE WHERE THE PROGRAM WAS
WHEN THE TRAP OR INTERRUPT OCCURED.

5.2.2 START (PROGRAM INITIALIZATION)

THIS ROUTINE INITIALIZES ALL PROGRAM FLAGS AND COUNTERS, TYPES THE PROGRAM TITLE MESSAGE, AND INPUTS THE VECTOR AND CONTROL REGISTER ADDRESSES OF THE DH11 TO BE TESTED.

5.2.3 BEGIN (PROGRAM START AND RESTART)

THIS ROUTINE IS ENTERED IMMEDIATELY AFTER "START" AND EACH TIME A PROGRAM PASS HAS BEEN COMPLETED. THE ROUTINE SETS UP THE PROCESSOR STACK AND STATUS WORD AND THEN TRANSFERS CONTROL TO THE TEST AT WHICH TESTING WILL BEGIN. IF SW01=0 WHEN THIS ROUTINE IS ENTERED TESTING WILL START AT T1 (TEST 1). IF SW01=1 WHEN THIS ROUTINE IS ENTERED, TESTING WILL START AT THE PC ENTERED FROM THE TELETYPE KEYBOARD.

5.2.4 EOP (END OF PASS)

THIS ROUTINE IS ENTERED ONCE PER PASS AFTER ALL TESTS HAVE BEEN COMPLETED. THIS ROUTINE TYPES THE MAINDEC IDENTIFICATION CODE OF THE PROGRAM, CLEARS ERROR FLAGS AND UPDATES THE PASS COUNT. IF THE PROGRAM WAS LOADED UNDER ACT11 OR DDP, THE ROUTINE CHECKS FOR RETURN TO THE ACT11 OR DDP MONITOR. IF THE PROGRAM IS NOT UNDER MONITOR CONTROL, THE ROUTINE TRANSFERS TO BEGIN.

5.2.5 SCOPER (SCOPE LOOP AND ITERATION HANDLER)

THIS ROUTINE IS ENTERED EACH TIME A TEST IS COMPLETED. THE ROUTINE CHECKS FOR THE FOLLOWING UPON ENTRY

1. IF SW10=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST IN SEQUENCE, AFTER CLEARING ERROR FLAGS.

2. IF SW11=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST SEQUENCE, AFTER CLEARING ERROR FLAGS.

3. IF SW14=1, THE ROUTINE WILL LOOP ON THE CURRENT TEST REGARDLESS OF THE ITERATION COUNT.

IF NONE OF THE ABOVE IS TRUE, THE ROUTINE WILL ADD 1 TO THE COUNT OF TEST ITERATIONS, AND COMPARE THIS VALUE TO THE NUMBER OF ITERATIONS THAT SHOULD BE PERFORMED. IF THESE NUMBERS ARE EQUAL, THE ROUTINE WILL TRANSFER TO THE NEXT TEST IN SEQUENCE. IF THE NUMBERS ARE NOT EQUAL, THE TEST CURRENTLY IN PROGRESS WILL BE REPEATED.

5.2.6 SCOP1R (FREEZE ON CURRENT DATA)

THE CALL TO THIS ROUTINE FOLLOWS IMMEDIATELY AFTER THE CALL TO THE ERROR HANDLER IN THOSE TESTS THAT HAVE VARIABLE PARAMETERS. THIS ROUTINE IS ALWAYS ENTERED IN THOSE TESTS, WHETHER OR NOT AN ERROR OCCURS. IF SW09=1, THE ROUTINE WILL TRANSFER CONTROL BACK TO THE TEST AT A POINT WHICH WILL ALLOW REPEATING THE FUNCTION UNDER TEST CONTINUOUSLY WITH THE SAME DATA. IF THIS OPTION IS SELECTED, THE ROUTINE "SCOPER" IS NEVER ENTERED AND ITERATION COUNTS WILL NOT BE UPDATED.

5.2.7 ERRORS (ERROR HANDLER)

THIS ROUTINE IS ENTERED UPON ERROR DETECTION ONLY. WITH ALL CONSOLE SWITCHES DOWN, THE ROUTINE PROCEDES AS FOLLOWS:

1. THE PC OF THE INSTRUCTION THAT CALLED THE ERROR HANDLER IS ACCESSED THRU THE STACK, AND THEN THE EMT INSTRUCTION ITSELF IS FETCHED. THE 8 LSB OF THE EMT INSTRUCTION ARE THE ERROR CODE. THIS CODE IS USED TO ACCESS A TABLE OF ERROR MESSAGES AND ERROR DATA STORAGE LOCATIONS.

2. IF THE TEST THAT FAILED DID NOT FAIL PREVIOUSLY DURING THIS PASS, A COMPLETE ERROR REPORT IS MADE IF THE TEST THAT FAILED FAILED MORE THAN ONCE DURING THE CURRENT PASS, ONLY THE DATA RELATING TO THE FAILURE IS TYPED. IF SW13=1, NO ERROR TYPEOUT IS MADE.

3. THE ROUTINE NOW CHECKS FOR HALT ON ERROR. IF SW15=1 THE PROGRAM WILL HALT WITH THE PC OF THE CALL TO THE ERROR ROUTINE IN R0. IF SW15=0, THE PROGRAM WILL NOT HALT, BUT WILL CHECK FOR ESCAPE TO NEXT TEST.

4. IF SW10=0, THE ROUTINE WILL RETURN TO THE TEST IN PROGRESS. IF SW10=1, THE ROUTINE WILL ABORT THE CURRENT TEST, AND TRANSFER TO THE NEXT TEST IN SEQUENCE, THRU THE ROUTINE "SCOPER".

5.2.8 TRPSRV (TRAP DECODE AND DISPATCH)

THIS ROUTINE DECODES THE 8 LSB OF THE TRAP INSTRUCTION THAT CAUSED THE PROGRAM INTERRUPT, AND TRANSFERS CONTROL TO THE ROUTINE THRU THE TABLE "TRPTAB" USING THE 8 LSB OF THE TRAP INSTRUCTION AS AN OFFSET TO THE POINTER TO THE ROUTINE TO BE ENTERED.

5.3 PROGRAM AND OR OPERATOR ACTION

5.3.1 PROGRAM START WITH ALL SWITCHES DOWN

5.3.1.1 REFER TO SECTIONS 4.3.1 AND 4.3.2 FOR INITIAL PROGRAM BEHAVIOR.

5.3.1.2 AFTER "R" HAS BEEN TYPED BY THE PROGRAM, TEST EXECUTION WILL BEGIN. EACH TEST WILL BE REPEATED A SELECTED NUMBER OF ITERATIONS (SEE LISTING FOR EXACT NUMBER FOR EACH TEST) AND THEN THE PROGRAM WILL PROCEED TO THE NEXT TEST.

5.3.1.3 WHEN ALL ITERATIONS HAVE BEEN COMPLETED, THE PROGRAM WILL TYPE "CZDHH-C" AND THEN RESTART TESTING AT TEST 1 (LOCATION T1 IN THE PROGRAM).

5.3.1.4 IF AN ERROR OCCURS, THE PROGRAM WILL TYPE AN APPROPRIATE ERROR MESSAGE, AND THEN CONTINUE THE TEST IN PROGRESS.

5.3.2 PROGRAM START WITH SW00=1

THE PROGRAM WILL PERFORM AS DESCRIBED IN 4.3.1 AND 5.3.1

5.3.3 PROGRAM START WITH SW01=1

5.3.3.1 REFER TO SECTION 4.3.4 FOR INITIAL PROGRAM BEHAVIOR

5.3.3.2 TEST EXECUTION WILL START AT THE ADDRESS SPECIFIED AND WILL CONTINUE AS DESCRIBED IN 5.3.1.2

5.3.3.3 AFTER "CZDHH-C" HAS BEEN TYPED, THE PROGRAM WILL RESUME TESTING AT TEST 1

5.3.4 PROGRAM OPERATION WITH SW15=1

SAME AS 5.3.1, EXCEPT THAT IN THE CASE OF AN ERROR, THE PROGRAM WILL HALT AFTER THE ERROR TYPEOUT, AND THE PC+2 OF THE CALL TO THE ERROR ROUTINE WILL BE DISPLAYED IN R0.

5.3.5 PROGRAM OPERATION WITH SW13=1

SAME AS 5.3.1 EXCEPT THAT NO ERROR TYPEOUTS WILL OCCUR

5.3.6 PROGRAM OPERATION WITH SW11=1

SAME AS 5.3.1 EXCEPT THAT EACH TEST WILL BE REPEATED ONCE ONLY

5.3.7 PROGRAM OPERATION WITH SW10=1

SAME AS 5.3.1, EXCEPT THAT IN THE CASE OF AN ERROR THE CURRENT TEST WILL BE ABORTED, AND THE PROGRAM WILL PROCEED TO THE NEXT TEST IN SEQUENCE.

5.3.8 PROGRAM OPERATION WITH SW14=1, OR SW09=1

THESE FUNCTIONS ARE NORMALLY USED FOR TROUBLE SHOOTING. SEE SECTION 6.3 FOR THEIR USE.

```
                                6.0  ERRORS
                                6.1  ERROR HALTS

                                THE ERROR MESSAGE FORMAT FOR ALL ERROR TYPEOUTS IS AS FOLLOWS

                                PC+2                MESSAGE
                                                    HEADER (IF APPLICABLE)
                                                    DATA   (IF APPLICABLE)

                                WHERE
                                PC+2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER + 2;
                                MESSAGE IS AN ASCII MESSAGE DESCRIBING (BRIEFLY) THE FAILURE;
                                HEADER IS A DESCRIPTION OF THE DATA TO FOLLOW;
                                DATA IS OCTAL INFORMATION RELATING TO THE CAUSE OF  THE  FAILURE.   IF
```

THE SAME ERRROR OCCURS IN A GIVEN TEST ON THE SAME PASS, AND IF DATA IS ASSOCIATED WITH THAT ERROR, ONLY DATA IS TYPE ON SUCCEEDING ERROR TYPEOUTS.

IF NO DATA IS ASSOCIATED WITH THE ERROR THE COMPLETE ERROR MESSAGE IS TYPED.

6.1.1 ERROR DESCRIPTIONS

SEE LISTING FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15=0

IF THE PROGRAM IS RUN WITH SW15=0, NO OPERATOR ACTION IS REQUIRED TO CONTINUE TESTING

6.2.2 SW15=1

IF THE PROGRAM IS RUN WITH SW15=1, TO CONTINUE TESTING AFTER THE PROGRAM HAS HALTED, PRESS THE PROCESSOR CONSOLE CONTINUE SWITCH

6.2.3 ILLEGAL INTERRUPTS

IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN THE TRAPCATCHER. THE ADDRESS AT WHICH THE PROGRAM HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT OCCURED. THE PROGRAM MUST BE RESTARTED AT 200 TO RECOVER FROM THIS ERROR.

6.3 SCOPE LOOPING

6.3.1 TO SCOPE ON A SPECIFIC TEST, SET SW14=1 AND SW13=1. THIS WILL CAUSE THE PROGRAM TO CONTINUOUSLY LOOP ON THE SAME TEST, AND WILL CAUSE ALL ERROR TYPEOUTS TO BE INHIBITED

6.3.2 TO SCOPE ON A SPECIFIC VALUE OF A PATAMETER WITHIN A TEST, SET SW09=1 TO FREEZE THE DATA. (SEE LISTING FOR THOSE TESTS THAT INCORPORATE THIS FEATURE)

6.3.3 PROGRAM START TO SCOPE LOOP ON SELECTED TEST

PERFORM SECTION 4.3.4 WITH SW14=1

7.0 RESTRICTIONS

7.1 STARTING

THE DH11 TEST CARD MUST BE INSTALLED

7.2 RUNNING

NONE

```
                                8.0  MISCELLANEOUS


                                8.1  EXECUTION TIME

                                THE TIME FOR ONE PASS OF THE PROGRAM (END OF TYPEOUT OF CZDHH-C  TO  END
                                OF  TYPEOUT  OF  CZDHH-C)  IS  GIVEN FOR VARIOUS PROCESSORS IN THE TABLE
                                BELOW

                                                    TIME

                                PROCESSOR

                                PDP-11/05,10

                                PDP-11/20

                                PDP-11/40

                                PDP-11/45
```

9.0 PROGRAM DESCRIPTION

FIRST, VERIFY THAT AUTO-ECHO WORKS ON ALL LINES BY TRANSMITTING ONE CHARACTER WITH AUTO ECHO ENABLED.

THEN A BINARY PATTERN IS TRANSMITTED ON ALL LINES EXCEPT THE ON WITH AUTO-ECHO ENABLED. A SINGLE CHARACTER IS TRANSMITTED ON THAT LINE. ALL DATA IS VERIFIED TO BE CORRECT.

10.0 LISTING

```
        !
```

```
      1                         ; DHMAC-A - DH11 MACRO LIBRARY
      2                         ; COPYRIGHT 1985, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
      3
      4
      5                         .LIST   ME
      6                         .NLIST  MC,MD,CND
      7
    104
    119
    131
    148
    158
    167
    303
    339
    373
    520                                                                                 ; 3
    563
    595
    607
    652
    664
    691
    712
    743                         ;  CMS REPLACEMENT HISTORY
    744
    745
    746                         ; *9 SKONETSKI 26-APR-1985 16:23:08 "FIXED TYPO CAUSING ASSEMBLY ERRORS"
    747                         ; *8 SKONETSKI 22-APR-1985 16:48:03 "TYPO ERROR IN VECTOR CHANGE CODE SOURCE FIXED"
    748                         ; *7 SKONETSKI 22-APR-1985 16:26:04 "ADDED CODE TO SET VECTORS FOR PWR FAIL, ERRORS, AND EMT
TRAPS."
    749                         ; *6 SKONETSKI 22-APR-1985 14:22:35 "FIXED BRANCH ERROR IN END OF PASS ROUTINE"
    750                         ; *5 SKONETSKI 22-APR-1985 08:28:54 "FIXED BUG (AN OCTASC MACRO CALL WAS WRONG) AND ADDED A
CLEAN END OF PASS
MESSAGE."
    751                         ; *4 SKONETSKI 18-APR-1985 14:20:15 "ADDED SOFTWARE SWITCH REG SUPPORT, BUT UNTESTED"
    752                         ; *3 SKONETSKI 12-APR-1985 10:34:52 "FIXED PROBLEMS WITH SPURIOUS CR/LFS"
    753                         ; *2 SKONETSKI 11-APR-1985 16:00:24 "ADDED MACRO FROM SYSMAC.SML THAT SIZES FOR SOFTWARE SWI
TCH REGISTER"
    754                         ; *1 SKONETSKI 11-APR-1985 15:49:05 "LIBRARY FOR DH11 DIAGNOSTICS"
```

```
                                .LIST   ME
                                .NLIST  MC,MD,CND
000000                          .HEADER /1972,1985/,/DH11 AUTO-ECHO TEST/,/CZDHH-C0/

                                ;STARTING PROCEDURE
                                ;LOAD PROGRAM
                                ;LOAD ADDRESS 000200
                                ;PRESS START
                                ;PROGRAM WILL TYPE DH11 AUTO-ECHO TEST
                                ;PROGRAM WILL TYPE "VECTOR ADDRESS-"
                                ;TYPE IN THE ADDRESS OF THE RECEIVER INTERRUPT VECTOR
                                ;FOR THE DH11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>
                                ;PROGRAM WILL TYPE "CONTROL REGISTER ADDRESS-"
                                ;TYPE IN THE ADDRESS OF THE SYSTEM CONTROL REGISTER
                                ;FOR THE DH11 TO BE TESTED, FOLLOWED BY <CARRIAGE RETURN>          ; 3
                                ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                                ;AT THE END OF A PASS, PROGRAM WILL TYPE " CZDHH-C0 "
                                ;AND THEN RESUM TESTING

                                .TITLE  CZDHH-C0
000000                          .ENABLE ABS
                                .NLIST  MC,MD,CND
                                .LIST   ME
000000                          .SYMBOLS

                                ;SWITCH REGISTER OPTIONS

                100000          SW15=100000             ;=1,HALT ON ERROR
                040000          SW14=40000              ;=1,LOOP ON CURRENT TEST
                020000          SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
                010000          SW12=10000
                004000          SW11=4000               ;=1,INHIBIT ITERATIONS
                002000          SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR          ; 3
                001000          SW09=1000               ;=1,LOOP WITH CURRENT DATA
                000400          SW08=400
                000100          SW06=100
                000040          SW05=40
                000020          SW04=20
                000010          SW03=10
                000004          SW02=4
                000002          SW01=2                  ;RESTART PROGRAM AT SELECTED TEST
                000001          SW00=1                  ;RESELECT VECTOR AND CONTROL REGISTER
                                                        ;ADDRESS AFTER PROGRAM RESTART
```

```
0

                        ;REGISTER DEFINITIONS

000000                  R0=%0           ;GENERAL REGISTER
000001                  R1=%1           ;GENERAL REGISTER
000002                  R2=%2           ;GENERAL REGISTER
000003                  R3=%3           ;GENERAL REGISTER
000004                  R4=%4           ;GENERAL REGISTER
000005                  R5=%5           ;GENERAL REGISTER
000006                  SP=%6           ;PROCESSOR STACK POINTER
000007                  PC=%7           ;PROGRAM COUNTER

                        ;LOCATION EQUIVALENCIES                                 ; 3

                        ;SWR=177570     ;CONSOLE SWITCH REGISTER                ; 4
                        ;LIGHTS=177570  ;PDP-11/45 DISPLAY REGISTER             ; 4
177776                  PS=177776       ;PROCESSOR STATUS WORD
015644                  STACK=ENDCOD+200    ;START OF PROCESSOR STACK           ; 3

                        ;INSTRUCTION DEFINITIONS

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                  PUSHR0=10046    ;SAVE R0 ON STACK
012600                  POPR0=12600     ;RESTORE R0 FROM STACK
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                  POP2SP=22626    ;INCREMENT STACK TWICE
                        ;
                        .MACRO  HLT     $A
                                EMT     $A
                        .ENDM   HLT
                        ;
                        ;
100000                  BIT15=100000
040000                  BIT14=40000                                             ; 3
020000                  BIT13=20000
010000                  BIT12=10000
004000                  BIT11=4000
002000                  BIT10=2000
001000                  BIT09=1000
000400                  BIT08=400
000200                  BIT07=200
000100                  BIT06=100
000040                  BIT05=40
000020                  BIT04=20
000010                  BIT03=10
000004                  BIT02=4
000002                  BIT01=2
000001                  BIT00=1
1 000000                .CATCH
```

```
0                                       ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
                000000          .=0
                000200          .REPT 200
                                        .+2             ;UNEXPECTED TRAP TO THIS LOCATION
                                        HALT            ;EXAMINE STACK TO FIND CAUSE
                                .ENDR
        000000  000002                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000002  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000004  000006                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000006  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000010  000012                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000012  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000014  000016                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000016  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000020  000022                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000022  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000024  000026                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000026  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000030  000032                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000032  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000034  000036                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000036  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000040  000042                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000042  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000044  000046                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000046  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000050  000052                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000052  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000054  000056                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000056  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000060  000062                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000062  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000064  000066                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000066  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000070  000072                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000072  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000074  000076                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000076  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000100  000102                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000102  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000104  000106                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000106  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000110  000112                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000112  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000114  000116                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000116  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000120  000122                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000122  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000124  000126                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000126  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000130  000132                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000132  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000134  000136                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000136  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000140  000142                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
        000142  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
        000144  000146                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
```

```
000146  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000150  000152          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000152  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000154  000156          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000156  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000160  000162          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000162  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000164  000166          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000166  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000170  000172          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000172  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000174  000176          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000176  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000200  000202          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000202  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000204  000206          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000206  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000210  000212          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000212  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000214  000216          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000216  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000220  000222          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000222  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000224  000226          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000226  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000230  000232          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000232  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000234  000236          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000236  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000240  000242          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000242  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000244  000246          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000246  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000250  000252          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000252  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000254  000256          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000256  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000260  000262          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000262  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000264  000266          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000266  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000270  000272          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000272  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000274  000276          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000276  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000300  000302          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000302  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000304  000306          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000306  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000310  000312          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000312  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000314  000316          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000316  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000320  000322          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000322  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
000324  000326          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000326  000000          HALT            ;EXAMINE STACK TO FIND CAUSE
```

```
000330  000332                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000332  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000334  000336                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000336  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000340  000342                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000342  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000344  000346                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000346  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000350  000352                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000352  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000354  000356                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000356  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000360  000362                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000362  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000364  000366                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000366  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000370  000372                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000372  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000374  000376                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000376  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000400  000402                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000402  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000404  000406                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000406  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000410  000412                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000412  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000414  000416                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000416  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000420  000422                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000422  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000424  000426                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000426  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000430  000432                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000432  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000434  000436                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000436  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000440  000442                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000442  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000444  000446                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000446  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000450  000452                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000452  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000454  000456                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000456  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000460  000462                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000462  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000464  000466                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000466  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000470  000472                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000472  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000474  000476                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000476  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000500  000502                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000502  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000504  000506                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000506  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000510  000512                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
```

```
000512  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000514  000516                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000516  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000520  000522                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000522  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000524  000526                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000526  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000530  000532                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000532  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000534  000536                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000536  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000540  000542                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000542  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000544  000546                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000546  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000550  000552                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000552  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000554  000556                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000556  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000560  000562                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000562  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000564  000566                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000566  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000570  000572                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000572  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000574  000576                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000576  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000600  000602                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000602  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000604  000606                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000606  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000610  000612                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000612  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000614  000616                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000616  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000620  000622                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000622  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000624  000626                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000626  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000630  000632                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000632  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000634  000636                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000636  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000640  000642                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000642  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000644  000646                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000646  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000650  000652                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000652  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000654  000656                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000656  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000660  000662                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000662  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000664  000666                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000666  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000670  000672                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000672  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
```

```
000674  000676                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000676  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000700  000702                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000702  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000704  000706                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000706  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000710  000712                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000712  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000714  000716                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000716  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000720  000722                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000722  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000724  000726                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000726  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000730  000732                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000732  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000734  000736                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000736  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000740  000742                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000742  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000744  000746                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000746  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000750  000752                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000752  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000754  000756                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000756  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000760  000762                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000762  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000764  000766                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000766  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000770  000772                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000772  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
000774  000776                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000776  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
  1 001000                      .SETVEC
```

```
0                                       ;STANDARD INTERRUPT VECTORS
                 000200                 .=200
  000200  000167  000600                JMP     START           ;GO TO START OF PROGRAM

1 000204                                .TRPDEF

                                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                        ;POINTERS TO SUBROUTINES CAN BE FOUND STARTING
                                        ;AT LOCATION "TRPTAB"

  000204                                TRPDEF  SCOPE,†/SCOPE LOOP AND ITERATION HANDLER/
                 104400                         SCOPE=TRAP+Y            ;SCOPE LOOP AND ITERATION HANDLER
                 000001                         Y=Y+1
  000204                                TRPDEF  TYPE,†/TELETYPE OUTPUT ROUTINE/
                 104401                         TYPE=TRAP+Y             ;TELETYPE OUTPUT ROUTINE
                 000002                         Y=Y+1
  000204                                TRPDEF  OCTASC,†/OCTAL TO ASCII CONVERSION/
                 104402                         OCTASC=TRAP+Y           ;OCTAL TO ASCII CONVERSION
                 000003                         Y=Y+1
  000204                                TRPDEF  INSTR,†/INPUT ASCII STRING/
                 104403                         INSTR=TRAP+Y            ;INPUT ASCII STRING
                 000004                         Y=Y+1
  000204                                TRPDEF  INSTER,†/STRING INPUT ERROR/
                 104404                         INSTER=TRAP+Y           ;STRING INPUT ERROR
                 000005                         Y=Y+1
  000204                                TRPDEF  PARAM,†/CONVERT STRING TO OCTAL, CHECK LIMITS/
                 104405                         PARAM=TRAP+Y            ;CONVERT STRING TO OCTAL, CHECK LIMITS
                 000006                         Y=Y+1
  000204                                TRPDEF  SAV05P,†/SAVE R0-R5, PC/
                 104406                         SAV05P=TRAP+Y           ;SAVE R0-R5, PC
                 000007                         Y=Y+1
  000204                                TRPDEF  RES05,†/RESTORE R0-R5/
                 104407                         RES05=TRAP+Y            ;RESTORE R0-R5
                 000010                         Y=Y+1
  000204                                TRPDEF  SCOPE1,†/CHECK FOR FREEZE ON CURRENT DATA/
                 104410                         SCOPE1=TRAP+Y           ;CHECK FOR FREEZE ON CURRENT DATA
                 000011                         Y=Y+1
2                                       .MACRO  CODEM1
3                                               MOV     DHSSR,DHSLR             ;SET UP ADDRESS OF SILO
4                                               INC     DHSLR                   ;STATUS REGISTER HIGH BYTE
5                                       .ENDM   CODEM1
6 000204                                .START  DHRVEC,3,4,DHSCR,0,177776,7,10,,,1
```

```
0       001000                  .=1000

                                ;PROGRAM INITIALIZATION
                                ;LOCK OUT INTERRUPTS
                                ;SET UP PROCESSOR STACK
                                ;SET UP POWER FAIL VECTOR
                                ;CLEAR PROGRAM FLAGS AND COUNTS
                                ;TYPE TITLE MESSAGE
                        .IIF NB <>,     ;DETERMINE MEMORY SIZE
                        .IIF NB <>,     ;SET UP TRACE TRAP RETURN

001000  177570          SWR:    .WORD   177570          ; SWITCH DHSCR ADDRESS          ; 4
001002  177570          LIGHTS: .WORD   177570          ; LIGHTS                        ; 4
                                                                                        ; 4
001004  012767  000340  176764  START:  MOV     #340,PS         ;LOCK OUT INTERRUPTS
001012  012706  015644          MOV     #STACK,SP       ;SET UP PROCESSOR STACK
001016  012702  000024          MOV     #24,R2          ; POINT TO VECTOR AREA          ; 7
001022  012722  014510          MOV     #PFAIL,(R2)+    ;SET UP POWER FAIL TRAP         ; 7
001026  012722  000340          MOV     #340, (R2)+     ;SERVICE AT LEVEL 7             ; 7
001032  012722  012350          MOV     #ERRORS, (R2)+  ;ERROR HANDLER                  ; 7
001036  012722  000340          MOV     #340, (R2)+     ;SERVICE AT LEVEL 7             ; 7
001042  012722  012562          MOV     #TRPSRV, (R2)+  ;GENERAL HANDLER DISPATCH SERVICE ; 7
001046  012712  000340          MOV     #340, (R2)      ;SERVICE AT LEVEL 7             ; 8
001052  005067  012556          CLR     STFLG           ;CLEAR TEST START FLAG
001056  005067  012512          CLR     PASCNT          ;CLEAR PASS COUNT
001062  005067  012510          CLR     ERRCNT          ;CLEAR ERROR COUNT
001066  005067  012500          CLR     ERRFLG          ;CLEAR ERROR FLAG
001072  005067  012474          CLR     ERRFLG          ;CLEAR LAST ERROR PC
001076  016746  176702          MOV     4, -(SP)        ; PUSH TRAP VECTOR              ; 4
001102  016746  176700          MOV     6, -(SP)                                        ; 4
001106  012767  001122  176670  MOV     #1$, 4          ; SET UP TRAP VECTOR            ; 4
001114  005777  177660          TST     @SWR            ; TEST SWITCH REGISTER ADDRESS  ; 4
001120  000405                 BR      2$              ; IF SUCCESSFUL, LEAVE IT ALONE ; 4
001122                  1$:                                                             ; 4
001122  012767  000176  177650  MOV     #176, SWR       ; POINT TO SOFT SWITCH DHSCR    ; 4
001130  005067  177646          CLR     LIGHTS          ; 0 MEANS WE ARE NOT GOING TO USE LIGHTS ; 4
001134                  2$:                                                             ; 5
001134  005726                 TST     (SP)+           ; CLEAN UP STACK                ; 4
001136  005726                 TST     (SP)+                                           ; 4
001140  012667  176642          MOV     (SP)+, 6                                        ; 4
001144  012667  176634          MOV     (SP)+, 4                                        ; 4
001150  104401  014660          TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
001154  005767  012452          TST     INIFLG          ;CHECK INITIALIZATION FLAG
                        .IF NB  <DHRVEC>
001160  001001                 BNE     VEC1            ;IF NOT 0, CHECK SWITCHES
                                                        ;FOR REINITIALIZATION
                        .IFF
                                BNE     BEGIN           ;IF NOT 0, START TEST
                        .ENDC
                        .IF NB  <>
                        SIZE:   CLR     R0
                                MOV     #2$,@#4         ;SET UP TIME OUT RETURN
                        1$:     TST     (R0)+           ;WILL TRAP WHEN NO MEMORY       ; 9
                                BR      1$              ;LOCATION RESPONDED, CONTINUE
                        2$:     MOV     R0,HCORE        ;R0 CONTAINS ADDRESS OF
                                SUB     #2,HCORE        ;NON EXISTANT MEMORY            ; 9
                                MOV     #6,@#4          ;RESTORE TRAPCATCHER
```

```
                                .ENDC
                                .IF NB  <>
                        TRACER: MOV     #1$,@#10                ;SET UP ILLEGAL INSTRUCTION TRAP RETURN
                                SXT     R0                      ;DO 11/40, 11/45 INSTRUCTION
                                MOV     #RTT,TRTRET             ;11/40,45 RTT RETURN FROM TRACE TRAP
                                BR      2$
                        1$:     MOV     #RTI,TRTRET             ;1105,10,20 RTI RETURN FROM TRACE TRAP
                                MOV     #12,@#10                ;RESTROE TRAPCATCHER
                                MOV     #TRTRET,@#16            ;SET UP TRACE TRAP VECTOR
                                .ENDC
                                .IF NB  <DHRVEC>                                                        ; 3
                                .IF B   <>
001162  000404                  BR      VEC2
                                .IFF
                                TST     INIFLG                  ;IF INITIALIZE FLAG=0
                                BEQ     VEC2                    ;GET VECTOR AND CSR ADDRESS
                                .ENDC
001164  032777  000001  177606  VEC1:   BIT     #SW00,@#SWR             ;IF SW00=1, GET NEW VECTOR      ; 4
001172  001445                  BEQ     BEGIN                   ;AND CSR
001174                          VEC2:                                                                   ; 4
001174  012701  000300          MOV     #300,R1                                                         ; 4
001200  012702  000302          MOV     #302,R2
001204  012703  000004          MOV     #4,R3
001210  010211                  1$:     MOV     R2,(R1)                 ;RESTORE TRAPCATCHER
001212  005012                  CLR     (R2)                    ;IN FLOATING VECTOR AREA
001214  060301                  ADD     R3,R1
001216  060302                  ADD     R3,R2
001220  020127  001000          CMP     R1,#1000
001224  001371                  BNE     1$
001226  104403                  INSTR                           ;INPUT ADDRESS OF DEVICE VECTOR
001230  014712                  MVECTOR                         ;MESSAGE "VECTOR ADDRESS-"
001232  104405                  PARAM                           ;CONVERT STRING TO OCTAL
001234  000300                  300                             ;LOW LIMIT
001236  000770                  770                             ;HIGH  LIMIT                    ; 3
001240  013562                  DHRVEC                          ;LOCATIONS TO BE FILLED
001242     003                  .BYTE   3                       ;NUMBER OF LOCATIONS
001243     004                  .BYTE   4                       ;LSB MASK
001244  104403                  INSTR                           ;INPUT ADDRESS OF DEVICE CSR
001246  014734                  MREGAD                          ;MESSAGE "CONTROL REGISTER ADDRESS-"
001250  104405                  PARAM                           ;CONVERT STRING TO OCTAL
001252  000000                  0                               ;LOW LIMIT
001254  177776                  177776                          ;HIGH LIMIT
001256  013540                  DHSCR                           ;LOCATIONS TO BE FILLED
001260     007                  .BYTE   7                       ;NUMBER OF LOCATIONS
001261     010                  .BYTE   10                      ;LSB MASK
                                .ENDC
                                .IF NB  <1>
001262                          CODEM1
001262  016767  012270  012270          MOV     DHSSR,DHSLR             ;SET UP ADDRESS OF SILO
001270  005267  012264                  INC     DHSLR                   ;STATUS REGISTER HIGH BYTE
                                .ENDC
001274  005767  012332          TST     INIFLG                  ;IF INITIALIZATION FLAG
001300  001002                  BNE     BEGIN                   ;IS CLEARED
001302  005167  012324          COM     INIFLG                  ;SET IT

                                ;PROGRAM START                                                          ; 3
                                ;CHECK FOR PROGRAM START AT SELECTED ADDRESS
```

```
001306  012767  000340  176462  BEGIN:  MOV     @340,PS                 ;LOCK OUT INTERRUPTS
001314  012706  015644                  MOV     @STACK,SP               ;SET UP PROCESSOR STACK
001320  032777  000002  177452          BIT     @SW01,@SWR              ;IF SW01=1                      ; 4
001326  001410                          BEQ     1$                      ;GET PC FOR PROGRAM START
001330  104403                          INSTR                           ;GET PC
001332  015123                          MTSTPC                          ;MESSAGE "TEST PC"
001334  104405                          PARAM                           ;CONVERT STRING TO OCTAL
001336  000000                          0
001340  017500                          17500
001342  013600                          RETRN
001344     001                  .BYTE   1
001345     001                  .BYTE   1
001346  000410                          BR      2$
001350  012767  001400  012222  1$:     MOV     @T1,RETRN               ;NORMAL START, TEST 1
001356  005767  012252                  TST     STFLG                   ;IF LOOPING, BYPASS TYPEOUT
001362  001004                          BNE     3$
001364  005167  012244                  COM     STFLG
001370  104401  015117          2$:     TYPE    ,MR                     ;TYPE "R" TO INDICATE START
001374  000177  012200          3$:     JMP     @RETRN                  ;START TESTING                  ; 3
```

```
                        .MACRO  AUTO1   XLINE,XBIT,K

                                ;ENABLE AUTO ECHO ON LINE 'XLINE'
                                ;TRANSMIT ONE CHARACTER ON LINE 'XLINE'
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

                        TS \XN,100,4$
                                MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
                        1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                ;BUS ADDRESS REGISTERS
                                MOV     #'XLINE',@DHSCR         ;SELECT LINE 'XLINE'
                                MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
                                MOV     #TWRD'K',@DHBA          ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
                                MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
                                CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
                                MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                ;8 BITS PER CHARACTER,
                                                                ;AUTO ECHO ENABLED ON LINE 'XLINE'
                                MOV     #'XBIT',@DHBAR          ;SET BAR BIT FOR LINE 'K'
                        2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
                                BPL     2$                      ;BE RECEIVED
                                INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
                                MOV     @DMNRC,R4               ;READ CHARACTER
                                CMP     R4,TWRD'K'              ;IS CHARACTER CORRECT
                                BEQ     3$
                                MOV     TWRD'K',R5              ;(R5)=EXPECTED CHARACTER
                                CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
                                HLT     0                       ;CHARACTER ECHOED INCORRECTLY
                                BR      4$                      ;RESTART TEST
                        3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
                                BGT     2$                      ;BEEN RECEIVED, CONTINUE
                                BMI     4$
                                BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
                                BR      2$                      ;GET 1 MORE CHARACTER
                        4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
                        .ENDM   AUTO1
```

```
                        .MACRO  AUTO2   XLINE,XBIT,K

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 'K'
                                ;TRANSMIT 1 CHARACTER ON LINE 'K' WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

                        TS \XN,10,5$
                                MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
                                JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                ;400 (OCTAL) CHARACTERS
                                MOV     #'K',@DHSCR             ;SELECT LINE XLINE FOR TESTING
                                MOV     #TWRD'K',@DHBA  ;CHARACTER TO BE TRANSMITTED
                                                                ;ON LINE XLINE IN AUTO ECHO MODE
                                MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE XLINE
                                MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE XLINE
                                BIC     #'XBIT',LINACT  ;CLEAR LINE ACTIVE BIT
                                MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
                                CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
                        1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
                                BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
                                MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
                                SWAB    R3
                                BIC     #177760,R3              ;CLEAR STATUS BITS
                                MOV     R3,R2
                                ASL     R2
                                CMP     R3,#'K'         ;IF LINE NUMBER IS XLINE
                                BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
                                CMP     RBUF(R2),R4             ;IF NOT LINE XLINE, CHECK DATA
                                BEQ     2$
                                MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
                                HLT     1                       ;NON ECHOED DATA ERROR
                                BR      4$
                        2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
                                BNE     1$                      ;CONTINUE IF NOT DONE
                                BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
                        3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
                                BNE     1$                      ;EXIT
                                MOV     #'K',@DHSCR             ;SELECT LINE XLINE
                                BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE XLINE
                                TSTB    @DHSLR                  ;GET REST OF CHARACTERS
                                BNE     1$                      ;AND CHECK
                                BR      5$
                        4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
                                CMP     R4,TWRD'K'              ;CHECK ECHOED DATA
                                BEQ     3$
                                MOV     TWRD'K',R5                      ;(R5)=EXPECTED ECHOED DATA
                                HLT     ?                       ;ECHOED DATA ERROR
                        5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
                        .ENDM   AUTO2
```

```
                        .MACRO  SSETALL

                                ;SET BYTE COUNT FOR ALL LINES TO 400
                                ;SET BUS ADDRESS FOR ALL LINES TO TBUF
                                ;CLEAR EXPECTED CHARACTER BUFFERS
                                ;SET LINE ACTIVE BITS FOR ALL LINES

                        SETALL: MOV     #20,R0                  ;SET UP TO LOAD 16
                                                                ;BYTE COUNT AND BUS ADDRESS
                                                                ;MEMORY LOCATIONS
                                CLR     R1                      ;SET UP TO GENERATE EXPECTED
                                                                ;RECEIVED CHARACTER BUFFER
                                MOV     #200,R2                 ;WILL BE HIGH BYTE
                                                                ;OF EXPECTED RECEIVED CHARACTER
                                MOV     #1,R3                   ;OFFSET FOR HIGH BYTE
                        1$:     MOV     #TBUF,@DHBA             ;LOAD BUS ADDRESS
                                MOV     #-400,@DHBC             ;LOAD BYTE COUNT
                                MOV     #31403,@DHLPR           ;SET LINE SPEED TO 4800 BAUD
                                CLRB    RBUF(R1)
                                                                ;RECEIVED CHARACTER
                                MOVB    R2,RBUF(R3)             ;LOAD HIGH BYTE
                                INC     @DHSCR                  ;ADVANCE LINE NUMBER TO NEXT LINE
                                INC     R2                      ;UPDATE POINTERS
                                ADD     #2,R1
                                ADD     #2,R3
                                DEC     R0                      ;CONTINUE IF NOT DONE
                                BNE     1$
                                MOV     #-1,LINACT              ;SET ACTIVE FLAGS FOR ALL LINES
                                RTS     PC                      ;RETURN TO CALLING ROUTINE
                        .ENDM   SSETALL

                        .MACRO  CCLRALL

                                ;CLEAR ALL BYTE COUNT AND BUS ADDRESS REGISTERS

                        CLRALL: MOV     #20,R0                  ;SET UP TO CLEAR 16
                        1$:     CLR     @DHBA                   ;CLEAR BUS ADDRESS
                                CLR     @DHBC                   ;CLEAR BYTE COUNT
                                INC     @DHSCR                  ;ANVANCE LINE NUMBER
                                DEC     R0                      :CONTINUE IF NOT DONE
                                BNE     1$
                                RTS     PC                      ;RETURN TO CALLING ROUTINE
                        .ENDM   CCLRALL
```

```
                        XLINE=LINE
                        XBIT=BITX
                        K=KX
                        LINE=0
                        BITX=1
                        KX=0
                        .REPT   20
                        AUTO1   \LINE,\BITX,\KX
                        .NLIST
                        LINE=LINE+1
                        BITX=BITX+BITX
                        KX=KX+1
                        .LIST
                        .ENDR
001400                  AUTO1   \LINE,\BITX,\KX

                                ;ENABLE AUTO ECHO ON LINE 0
                                ;TRANSMIT ONE CHARACTER ON LINE 0
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

001400                  TS \XN,100,4$
001400  012767  000340  176370 T1:     MOV     #340,PS         ;DISABLE ALL INTERRUPTS
001406  012767  000100  012172         MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
001414  012767  001554  012160         MOV     #4$,.ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #.FREEZ1        ;SET UP TO LOOP WITH DATA       ; 3
                        .ENDC
                        XN=XN+1
001422  012777  004000  012110         MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
001430  004767  011744         1$:     JSR     PC,CLRALL       ;CLEAR ALL BYTE COUNT AN
                                                                ;BUS ADDRESS REGISTERS
001434  012777  000000  012076         MOV     #0,@DHSCR       ;SELECT LINE 0
001442  012777  177777  012100         MOV     #-1,@DHBC       ;SET BYTE COUNT TO 1
001450  012777  014310  012070         MOV     #TWRDO,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
001456  012700  000100                 MOV     #100,R0         ;SET UP TO RECEIVE 64 CHARACTERS
001462  005001                         CLR     R1              ;COUNT OF CHARACTERS RECEIVED
001464  012777  133503  012052         MOV     #133503,@DHLPR  ;SET UP SPEED FOR 9600 BAUD
                                                                ;8 BITS PER CHARACTER.
                                                                ;AUTO ECHO ENABLED ON LINE 0
001472  012777  000001  012052         MOV     #1,@DHBAR       ;SET BAR BIT FOR LINE 0
001500  105777  012034         2$:     TSTB    @DHSCR          ;WAIT FOR CHARACTER TO
001504  100375                         BPL     2$              ;BE RECEIVED
001506  005201                         INC     R1              ;UPDATE RECEIVED CHARACTER COUNT
001510  017704  012026                 MOV     @DHNRC,R4       ;READ CHARACTER
001514  020467  012570                 CMP     R4,TWRDO        ;IS CHARACTER CORRECT
001520  001406                         BEQ     3$
001522  016705  012562                 MOV     TWRDO,R5        ;(R5)=EXPECTED CHARACTER
001526  005077  012012                 CLR     @DHLPR          ;SHUT OFF AUTO ECHO
001532                                 HLT     0               ;CHARACTER ECHOED INCORRECTLY
001532  104000                         EMT     0
001534  000407                         BR      4$              ;RESTART TEST
001536  005300                 3$:     DEC     R0              ;IF 64 CHARACTERS HAVE NOT
001540  003357                         BGT     2$              ;BEEN RECEIVED, CONTINUE
001542  100404                         BMI     4$
```

```
001544  042777  100000  011772          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
001552  000752                          BR      2$                      ;GET 1 MORE CHARACTER
001554  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000001                  LINE=LINE+1
        000002                  BITX=BITX+BITX
        000001                  KX=KX+1
001556                          AUTO1   \LINE,\BITX,\KX

                                ;ENABLE AUTO ECHO ON LINE 1
                                ;TRANSMIT ONE CHARACTER ON LINE 1
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

001556                          TS \XN,100,4$
001556  012767  000340  176212  T2:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
001564  012767  000100  012014          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
001572  012767  001732  012002          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000003                  XN=XN+1
001600  012777  004000  011732          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
001606  004767  011566          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
001612  012777  000001  011720          MOV     #1,@DHSCR               ;SELECT LINE 1
001620  012777  177777  011722          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
001626  012777  014312  011712          MOV     #TWRD1,@DHBA            ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
001634  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
001640  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
001642  012777  133503  011674          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER.
                                                                        ;AUTO ECHO ENABLED ON LINE 1
001650  012777  000002  011674          MOV     #2,@DHBAR               ;SET BAR BIT FOR LINE 1
001656  105777  011656          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
001662  100375                          BPL     2$                      ;BE RECEIVED
001664  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
001666  017704  011650                  MOV     @DHNRC,R4               ;READ CHARACTER
001672  020467  012414                  CMP     R4,TWRD1                ;IS CHARACTER CORRECT
001676  001406                          BEQ     3$
001700  016705  012406                  MOV     TWRD1,R5                ;(R5)=EXPECTED CHARACTER
001704  005077  011634                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
001710                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
001710  104000                          EMT     0
001712  000407                          BR      4$                      ;RESTART TEST
001714  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
001716  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
001720  100404                          BMI     4$
001722  042777  100000  011614          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
001730  000752                          BR      2$                      ;GET 1 MORE CHARACTER
001732  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000002                  LINE=LINE+1
        000004                  BITX=BITX+BITX
        000002                  KX=KX+1
001734                          AUTO1   \LINE,\BITX,\KX
```

```
                                        ;ENABLE AUTO ECHO ON LINE 2
                                        ;TRANSMIT ONE CHARACTER ON LINE 2
                                        ;AT 9600 BAUD, 8 BITS.
                                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

001734                                  TS \XN,100,4$
001734  012767  000340  176034  T3:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
001742  012767  000100  011636          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
001750  012767  002110  011624          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                        .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                        .ENDC
        000004                          XN=XN+1
001756  012777  004000  011554          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
001764  004767  011410          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
001770  012777  000002  011542          MOV     #2,@DHSCR               ;SELECT LINE 2
001776  012777  177777  011544          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
002004  012777  014314  011534          MOV     #TWRD2,@DHBA            ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002012  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
002016  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
002020  012777  133503  011516          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 2
002026  012777  000004  011516          MOV     #4,@DHBAR               ;SET BAR BIT FOR LINE 2
002034  105777  011500          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
002040  100375                          BPL     2$                      ;BE RECEIVED
002042  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
002044  017704  011472                  MOV     @DHNRC,R4               ;READ CHARACTER
002050  020467  012240                  CMP     R4,TWRD2                ;IS CHARACTER CORRECT
002054  001406                          BEQ     3$
002056  016705  012232                  MOV     TWRD2,R5                ;(R5)=EXPECTED CHARACTER
002062  005077  011456                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
002066                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
002066  104000                          EMT     0
002070  000407                          BR      4$                      ;RESTART TEST
002072  005300          3$:             DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
002074  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
002076  100404                          BMI     4$
002100  042777  100000  011436          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
002106  000752                          BR      2$                      ;GET 1 MORE CHARACTER
002110  104400          4$:             SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000003                          LINE=LINE+1
        000010                          BITX=BITX+BITX
        000003                          KX=KX+1
002112                          AUTO1   \LINE,\BITX,\KX

                                        ;ENABLE AUTO ECHO ON LINE 3
                                        ;TRANSMIT ONE CHARACTER ON LINE 3
                                        ;AT 9600 BAUD, 8 BITS.
                                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

002112                                  TS \XN,100,4$
```

```
002112  012767  000340  175656  T4:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
002120  012767  000100  011460          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
002126  012767  002266  011446          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000005                  XN=XN+1
002134  012777  004000  011376          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
002142  004767  011232          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
002146  012777  000003  011364          MOV     #3,@DHSCR               ;SELECT LINE 3
002154  012777  177777  011366          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
002162  012777  014316  011356          MOV     #TWRD3,@DHBA            ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002170  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
002174  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
002176  012777  133503  011340          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 3
002204  012777  000010  011340          MOV     #10,@DHBAR              ;SET BAR BIT FOR LINE 3
002212  105777  011322          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
002216  100375                          BPL     2$                      ;BE RECEIVED
002220  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
002222  017704  011314                  MOV     @DHNRC,R4               ;READ CHARACTER
002226  020467  012064                  CMP     R4,TWRD3                ;IS CHARACTER CORRECT
002232  001406                          BEQ     3$
002234  016705  012056                  MOV     TWRD3,R5                ;(R5)=EXPECTED CHARACTER
002240  005077  011300                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
002244                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
002244  104000                          EMT     0
002246  000407                          BR      4$                      ;RESTART TEST
002250  005300          3$:             DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
002252  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
002254  100404                          BMI     4$
002256  042777  100000  011260          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
002264  000752                          BR      2$                      ;GET 1 MORE CHARACTER
002266  104400          4$:             SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000004                  LINE=LINE+1
        000020                  BITX=BITX+BITX
        000004                  KX=KX+1
002270                          AUTO1   \LINE,\BITX,\KX

                                ;ENABLE AUTO ECHO ON LINE 4
                                ;TRANSMIT ONE CHARACTER ON LINE 4
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

002270                          TS \XN,100,4$
002270  012767  000340  175500  T5:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
002276  012767  000100  011302          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
002304  012767  002444  011270          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000006                  XN=XN+1
002312  012777  004000  011220          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
```

```
002320  004767  011054          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
002324  012777  000004  011206          MOV     #4,@DHSCR               ;SELECT LINE 4
002332  012777  177777  011210          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
002340  012777  014320  011200          MOV     #TWRD4,@DHBA            ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002346  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
002352  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
002354  012777  133503  011162          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 4
002362  012777  000020  011162          MOV     #20,@DHBAR              ;SET BAR BIT FOR LINE 4
002370  105777  011144          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
002374  100375                          BPL     2$                      ;BE RECEIVED
002376  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
002400  017704  011136                  MOV     @DHNRC,R4               ;READ CHARACTER
002404  020467  011710                  CMP     R4,TWRD4                ;IS CHARACTER CORRECT
002410  001406                          BEQ     3$
002412  016705  011702                  MOV     TWRD4,R5                ;(R5)=EXPECTED CHARACTER
002416  005077  011122                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
002422                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
002422  104000                          EMT     0
002424  000407                          BR      4$                      ;RESTART TEST
002426  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
002430  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
002432  100404                          BMI     4$
002434  042777  100000  011102          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
002442  000752                          BR      2$                      ;GET 1 MORE CHARACTER
002444  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000005                  LINE=LINE+1
        000040                  BITX=BITX+BITX
        000005                  KX=KX+1
002446                          AUTO1   \LINE,\BITX,\KX

                                        ;ENABLE AUTO ECHO ON LINE 5
                                        ;TRANSMIT ONE CHARACTER ON LINE 5
                                        ;AT 9600 BAUD, 8 BITS.
                                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

002446                          TS \XN,100,4$
002446  012767  000340  175322  T6:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
002454  012767  000100  011124          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
002462  012767  002622  011112          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #.,FREEZ1               ;SET UP TO LOOP WITH DATA          3
                                .ENDC
        000007                  XN=XN+1
002470  012777  004000  011042          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
002476  004767  010676          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
002502  012777  000005  011030          MOV     #5,@DHSCR               ;SELECT LINE 5
002510  012777  177777  011032          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
002516  012777  014322  011022          MOV     #TWRD5,@DHBA            ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002524  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
002530  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
002532  012777  133503  011004          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
```

```
                                                                ;8 BITS PER CHARACTER,
                                                                ;AUTO ECHO ENABLED ON LINE 5
002540  012777  000040  011004          MOV     #40,@DHBAR      ;SET BAR BIT FOR LINE 5
002546  105777  010766          2$:     TSTB    @DHSCR          ;WAIT FOR CHARACTER TO
002552  100375                  BPL     2$                      ;BE RECEIVED
002554  005201                  INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
002556  017704  010760          MOV     @DHNRC,R4               ;READ CHARACTER
002562  020467  011534          CMP     R4,TWRD5                ;IS CHARACTER CORRECT
002566  001406                  BEQ     3$
002570  016705  011526          MOV     TWRD5,R5                ;(R5)=EXPECTED CHARACTER
002574  005077  010744          CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
002600                          HLT     0                       ;CHARACTER ECHOED INCORRECTLY
002600  104000                  EMT     0
002602  000407                  BR      4$                      ;RESTART TEST
002604  005300          3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
002606  003357                  BGT     2$                      ;BEEN RECEIVED, CONTINUE
002610  100404                  BMI     4$
002612  042777  100000  010724        BIC     #100000,@DHLPR    ;SHUT OFF AUTO-ECHO
002620  000752                  BR      2$                      ;GET 1 MORE CHARACTER
002622  104400          4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000006          LINE=LINE+1
        000100          BITX=BITX+BITX
        000006          KX=KX+1
002624                  AUTO1   \LINE,\BITX,\KX

                        ;ENABLE AUTO ECHO ON LINE 6
                        ;TRANSMIT ONE CHARACTER ON LINE 6
                        ;AT 9600 BAUD, 8 BITS.
                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

002624                  TS \XN,100,4$
002624  012767  000340  175144  T7:     MOV     #340,PS         ;DISABLE ALL INTERRUPTS
002632  012767  000100  010746          MOV     #100,ICOUNT     ;SET UP FOR 100 ITERATIONS
002640  012767  003000  010734          MOV     #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                        .ENDC
        000010          XN=XN+1
002646  012777  004000  010664          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
002654  004767  010520          1$:     JSR     PC,CLRALL       ;CLEAR ALL BYTE COUNT AN
                                                                ;BUS ADDRESS REGISTERS
002660  012777  000006  010652          MOV     #6,@DHSCR       ;SELECT LINE 6
002666  012777  177777  010654          MOV     #-1,@DHBC       ;SET BYTE COUNT TO 1
002674  012777  014324  010644          MOV     #TWRD6,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002702  012700  000100                  MOV     #100,R0         ;SET UP TO RECEIVE 64 CHARACTERS
002706  005001                          CLR     R1              ;COUNT OF CHARACTERS RECEIVED
002710  012777  133503  010626          MOV     #133503,@DHLPR  ;SET UP SPEED FOR 9600 BAUD
                                                                ;8 BITS PER CHARACTER,
                                                                ;AUTO ECHO ENABLED ON LINE 6
002716  012777  000100  010626          MOV     #100,@DHBAR     ;SET BAR BIT FOR LINE 6
002724  105777  010610          2$:     TSTB    @DHSCR          ;WAIT FOR CHARACTER TO
002730  100375                  BPL     2$                      ;BE RECEIVED
002732  005201                  INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
002734  017704  010602          MOV     @DHNRC,R4               ;READ CHARACTER
002740  020467  011360          CMP     R4,TWRD6                ;IS CHARACTER CORRECT
```

```
002744  001406                          BEQ     3$
002746  016705  011352                  MOV     TWRD6,R5                ;(R5)=EXPECTED CHARACTER
002752  005077  010566                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
002756                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
002756  104000                          EMT     0
002760  000407                          BR      4$                      ;RESTART TEST
002762  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
002764  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
002766  100404                          BMI     4$
002770  042777  100000  010546          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
002776  000752                          BR      2$                      ;GET 1 MORE CHARACTER
003000  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000007                  LINE=LINE+1
        000200                  BITX=BITX+BITX
        000007                  KX=KX+1
003002                          AUTO1   \LINE,\BITX,\KX

                                ;ENABLE AUTO ECHO ON LINE 7
                                ;TRANSMIT ONE CHARACTER ON LINE 7
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

003002                          TS \XN,100,4$
003002  012767  000340  174766  T10:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
003010  012767  000100  010570          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
003016  012767  003156  010556          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000011                  XN=XN+1
003024  012777  004000  010506          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
003032  004767  010342          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
003036  012777  000007  010474          MOV     #7,@DHSCR               ;SELECT LINE 7
003044  012777  177777  010476          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
003052  012777  014326  010466          MOV     #TWRD7,@DHBA            ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003060  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
003064  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
003066  012777  133503  010450          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 7
003074  012777  000200  010450          MOV     #200,@DHBAR             ;SET BAR BIT FOR LINE 7
003102  105777  010432          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
003106  100375                          BPL     2$                      ;BE RECEIVED
003110  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
003112  017704  010424                  MOV     @DHNRC,R4               ;READ CHARACTER
003116  020467  011204                  CMP     R4,TWRD7                ;IS CHARACTER CORRECT
003122  001406                          BEQ     3$
003124  016705  011176                  MOV     TWRD7,R5                ;(R5)=EXPECTED CHARACTER
003130  005077  010410                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
003134                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
003134  104000                          EMT     0
003136  000407                          BR      4$                      ;RESTART TEST
003140  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
003142  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
```

```
003144  100404                          BMI     4$
003146  042777  100000  010370          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
003154  000752                          BR      2$                      ;GET 1 MORE CHARACTER
003156  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000010                  LINE=LINE+1
        000400                  BITX=BITX+BITX
        000010                  KX=KX+1
003160                          AUTO1   \LINE,\BITX,\KX

                                ;ENABLE AUTO ECHO ON LINE 10
                                ;TRANSMIT ONE CHARACTER ON LINE 10
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

003160                          TS \XN,100,4$
003160  012767  000340  174610  T11:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
003166  012767  000100  010412          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
003174  012767  003334  010400          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000012                  XN=XN+1
003202  012777  004000  010330          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
003210  004767  010164          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
003214  012777  000010  010316          MOV     #10,@DHSCR              ;SELECT LINE 10
003222  012777  177777  010320          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
003230  012777  014330  010310          MOV     #TWRD10,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003236  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
003242  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
003244  012777  133503  010272          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 10
003252  012777  000400  010272          MOV     #400,@DHBAR             ;SET BAR BIT FOR LINE 10
003260  105777  010254          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
003264  100375                          BPL     2$                      ;BE RECEIVED
003266  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
003270  017704  010246                  MOV     @DHNRC,R4               ;READ CHARACTER
003274  020467  011030                  CMP     R4,TWRD10               ;IS CHARACTER CORRECT
003300  001406                          BEQ     3$
003302  016705  011022                  MOV     TWRD10,R5               ;(R5)=EXPECTED CHARACTER
003306  005077  010232                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
003312                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
003312  104000                          EMT     0
003314  000407                          BR      4$                      ;RESTART TEST
003316  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
003320  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
003322  100404                          BMI     4$
003324  042777  100000  010212          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
003332  000752                          BR      2$                      ;GET 1 MORE CHARACTER
003334  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000011                  LINE=LINE+1
        001000                  BITX=BITX+BITX
        000011                  KX=KX+1
003336                          AUTO1   \LINE,\BITX,\KX
```

```
                                        ;ENABLE AUTO ECHO ON LINE 11
                                        ;TRANSMIT ONE CHARACTER ON LINE 11
                                        ;AT 9600 BAUD, 8 BITS.
                                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

003336                          TS \XN,100,4$
003336  012767  000340  174432  T12:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
003344  012767  000100  010234          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
003352  012767  003512  010222          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000013                  XN=XN+1
003360  012777  004000  010152          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
003366  004767  010006          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
003372  012777  000011  010140          MOV     #11,@DHSCR              ;SELECT LINE 11
003400  012777  177777  010142          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
003406  012777  014332  010132          MOV     #TWRD11,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003414  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
003420  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
003422  012777  133503  010114          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 11
003430  012777  001000  010114          MOV     #1000,@DHBAR            ;SET BAR BIT FOR LINE 11
003436  105777  010076          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
003442  100375                          BPL     2$                      ;BE RECEIVED
003444  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
003446  017704  010070                  MOV     @DHNRC,R4               ;READ CHARACTER
003452  020467  010654                  CMP     R4,TWRD11               ;IS CHARACTER CORRECT
003456  001406                          BEQ     3$
003460  016705  010646                  MOV     TWRD11,R5               ;(R5)=EXPECTED CHARACTER
003464  005077  010054                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
003470                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
003470  104000                          EMT     0
003472  000407                          BR      4$                      ;RESTART TEST
003474  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
003476  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
003500  100404                          BMI     4$
003502  042777  100000  010034          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
003510  000752                          BR      2$                      ;GET 1 MORE CHARACTER
003512  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000012                  LINE=LINE+1
        002000                  BITX=BITX+BITX
        000012                  KX=KX+1
003514                          AUTO1   \LINE,\BITX,\KX

                                        ;ENABLE AUTO ECHO ON LINE 12
                                        ;TRANSMIT ONE CHARACTER ON LINE 12
                                        ;AT 9600 BAUD, 8 BITS.
                                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.
```

```
003514                                  TS \XN,100,4$
003514  012767  000340  174254  T13:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
003522  012767  000100  010056          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
003530  012767  003670  010044          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                        .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                        .ENDC
        000014                          XN=XN+1
003536  012777  004000  007774          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
003544  004767  007630          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
003550  012777  000012  007762          MOV     #12,@DHSCR              ;SELECT LINE 12
003556  012777  177777  007764          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
003564  012777  014334  007754          MOV     #TWRD12,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003572  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
003576  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
003600  012777  133503  007736          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 12
003606  012777  002000  007736          MOV     #2000,@DHBAR            ;SET BAR BIT FOR LINE 12
003614  105777  007720          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
003620  100375                          BPL     2$                      ;BE RECEIVED
003622  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
003624  017704  007712                  MOV     @DHNRC,R4               ;READ CHARACTER
003630  020467  010500                  CMP     R4,TWRD12               ;IS CHARACTER CORRECT
003634  001406                          BEQ     3$
003636  016705  010472                  MOV     TWRD12,R5               ;(R5)=EXPECTED CHARACTER
003642  005077  007676                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
003646                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
003646  104000                          EMT     0
003650  000407                          BR      4$                      ;RESTART TEST
003652  005300          3$:             DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
003654  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
003656  100404                          BMI     4$
003660  042777  100000  007656          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
003666  000752                          BR      2$                      ;GET 1 MORE CHARACTER
003670  104400          4$:             SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000013                  LINE=LINE+1
        004000                  BITX=BITX+BITX
        000013                  KX=KX+1
003672                          AUTO1   \LINE,\BITX,\KX

                                        ;ENABLE AUTO ECHO ON LINE 13
                                        ;TRANSMIT ONE CHARACTER ON LINE 13
                                        ;AT 9600 BAUD, 8 BITS.
                                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

003672                                  TS \XN,100,4$
003672  012767  000340  174076  T14:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
003700  012767  000100  007700          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
003706  012767  004046  007666          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                        .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                        .ENDC
        000015                          XN=XN+1
```

```
003714  012777  004000  007616          MOV     @BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
003722  004767  007452          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
003726  012777  000013  007604          MOV     #13,@DHSCR              ;SELECT LINE 13
003734  012777  177777  007606          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
003742  012777  014336  007576          MOV     #TWRD13,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003750  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
003754  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
003756  012777  133503  007560          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 13
003764  012777  004000  007560          MOV     #4000,@DHBAR            ;SET BAR BIT FOR LINE 13
003772  105777  007542          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
003776  100375                          BPL     2$                      ;BE RECEIVED
004000  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
004002  017704  007534                  MOV     @DHNRC,R4               ;READ CHARACTER
004006  020467  010324                  CMP     R4,TWRD13               ;IS CHARACTER CORRECT
004012  001406                          BEQ     3$
004014  016705  010316                  MOV     TWRD13,R5               ;(R5)=EXPECTED CHARACTER
004020  005077  007520                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
004024                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
004024  104000                          EMT     0
004026  000407                          BR      4$                      ;RESTART TEST
004030  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
004032  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
004034  100404                          BMI     4$
004036  042777  100000  007500          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
004044  000752                          BR      2$                      ;GET 1 MORE CHARACTER
004046  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000014                  LINE=LINE+1
        010000                  BITX=BITX+BITX
        000014                  KX=KX+1
004050                          AUTO1   \LINE,\BITX,\KX

                                ;ENABLE AUTO ECHO ON LINE 14
                                ;TRANSMIT ONE CHARACTER ON LINE 14
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

004050                          TS \XN,100,4$
004050  012767  000340  173720  T15:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
004056  012767  000100  007522          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
004064  012767  004224  007510          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
        000016                  XN=XN+1
004072  012777  004000  007440          MOV     @BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
004100  004767  007274          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
004104  012777  000014  007426          MOV     #14,@DHSCR              ;SELECT LINE 14
004112  012777  177777  007430          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
004120  012777  014340  007420          MOV     #TWRD14,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004126  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
004132  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
```

```
004134  012777  133503  007402          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 14
004142  012777  010000  007402          MOV     #10000,@DHBAR           ;SET BAR BIT FOR LINE 14
004150  105777  007364          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
004154  100375                          BPL     2$                      ;BE RECEIVED
004156  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
004160  017704  007356                  MOV     @DHNRC,R4               ;READ CHARACTER
004164  020467  010150                  CMP     R4,TWRD14               ;IS CHARACTER CORRECT
004170  001406                          BEQ     3$
004172  016705  010142                  MOV     TWRD14,R5               ;(R5)=EXPECTED CHARACTER
004176  005077  007342                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
004202                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
004202  104000                          EMT     0
004204  000407                          BR      4$                      ;RESTART TEST
004206  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
004210  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
004212  100404                          BMI     4$
004214  042777  100000  007322          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
004222  000752                          BR      2$                      ;GET 1 MORE CHARACTER
004224  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000015                  LINE=LINE+1
        020000                  BITX=BITX+BITX
        000015                  KX=KX+1
004226                          AUTO1   \LINE,\BITX,\KX

                                        ;ENABLE AUTO ECHO ON LINE 15
                                        ;TRANSMIT ONE CHARACTER ON LINE 15
                                        ;AT 9600 BAUD, 8 BITS.
                                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

004226                          TS \XN,100,4$
004226  012767  000340  173542  T16:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
004234  012767  000100  007344          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
004242  012767  004402  007332          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000017                  XN=XN+1
004250  012777  004000  007262          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
004256  004767  007116          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
004262  012777  000015  007250          MOV     #15,@DHSCR              ;SELECT LINE 15
004270  012777  177777  007252          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
004276  012777  014342  007242          MOV     #TWRD15,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004304  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
004310  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
004312  012777  133503  007224          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 15
004320  012777  020000  007224          MOV     #20000,@DHBAR           ;SET BAR BIT FOR LINE 15
004326  105777  007206          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
004332  100375                          BPL     2$                      ;BE RECEIVED
004334  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
004336  017704  007200                  MOV     @DHNRC,R4               ;READ CHARACTER
```

```
004342  020467  007774          CMP     R4,TWRD15               ;IS CHARACTER CORRECT
004346  001406                  BEQ     3$
004350  016705  007766          MOV     TWRD15,R5               ;(R5)=EXPECTED CHARACTER
004354  005077  007164          CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
004360                          HLT     0                       ;CHARACTER ECHOED INCORRECTLY
004360  104000                  EMT     0
004362  000407                  BR      4$                      ;RESTART TEST
004364  005300          3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
004366  003357                  BGT     2$                      ;BEEN RECEIVED, CONTINUE
004370  100404                  BMI     4$
004372  042777  100000  007144  BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
004400  000752                  BR      2$                      ;GET 1 MORE CHARACTER
004402  104400          4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000016          LINE=LINE+1
        040000          BITX=BITX+BITX
        000016          KX=KX+1
004404                  AUTO1   \LINE,\BITX,\KX

                        ;ENABLE AUTO ECHO ON LINE 16
                        ;TRANSMIT ONE CHARACTER ON LINE 16
                        ;AT 9600 BAUD, 8 BITS.
                        ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                        ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                        ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

004404                  TS \XN,100,4$
004404  012767  000340  173364  T17:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
004412  012767  000100  007166          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
004420  012767  004560  007154          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                        .ENDC
        000020          XN=XN+1
004426  012777  004000  007104          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
004434  004767  006740          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
004440  012777  000016  007072          MOV     #16,@DHSCR              ;SELECT LINE 16
004446  012777  177777  007074          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
004454  012777  014344  007064          MOV     #TWRD16,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004462  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
004466  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
004470  012777  133503  007046          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 16
004476  012777  040000  007046          MOV     #40000,@DHBAR           ;SET BAR BIT FOR LINE 16
004504  105777  007030          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
004510  100375                  BPL     2$                      ;BE RECEIVED
004512  005201                  INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
004514  017704  007022          MOV     @DHNRC,R4               ;READ CHARACTER
004520  020467  007620          CMP     R4,TWRD16               ;IS CHARACTER CORRECT
004524  001406                  BEQ     3$
004526  016705  007612          MOV     TWRD16,R5               ;(R5)=EXPECTED CHARACTER
004532  005077  007006          CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
004536                          HLT     0                       ;CHARACTER ECHOED INCORRECTLY
004536  104000                  EMT     0
004540  000407                  BR      4$                      ;RESTART TEST
004542  005300          3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
```

```
004544  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
004546  100404                          BMI     4$
004550  042777  100000  006766          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
004556  000752                          BR      2$                      ;GET 1 MORE CHARACTER
004560  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000017                  LINE=LINE+1
        100000                  BITX=BITX+BITX
        000017                  KX=KX+1
004562                          AUTO1   \LINE,\BITX,\KX

                                ;ENABLE AUTO ECHO ON LINE 17
                                ;TRANSMIT ONE CHARACTER ON LINE 17
                                ;AT 9600 BAUD, 8 BITS.
                                ;RECEIVE AND VERIFY CHARACTERS UNTIL 64 HAVE BEEN RECEIVED.
                                ;AFTER 64 CHARACTERS HAVE BEEN RECEIVED, DISABLE AUTO ECHO.
                                ;EXACTLY ONE MORE CHARACTER SHOULD BE RECEIVED.

004562                          TS \XN,100,4$
004562  012767  000340  173206  T20:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
004570  012767  000100  007010          MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
004576  012767  004736  006776          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000021                  XN=XN+1
004604  012777  004000  006726          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
004612  004767  006562          1$:     JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AN
                                                                        ;BUS ADDRESS REGISTERS
004616  012777  000017  006714          MOV     #17,@DHSCR              ;SELECT LINE 17
004624  012777  177777  006716          MOV     #-1,@DHBC               ;SET BYTE COUNT TO 1
004632  012777  014346  006706          MOV     #TWRD17,@DHBA           ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004640  012700  000100                  MOV     #100,R0                 ;SET UP TO RECEIVE 64 CHARACTERS
004644  005001                          CLR     R1                      ;COUNT OF CHARACTERS RECEIVED
004646  012777  133503  006670          MOV     #133503,@DHLPR          ;SET UP SPEED FOR 9600 BAUD
                                                                        ;8 BITS PER CHARACTER,
                                                                        ;AUTO ECHO ENABLED ON LINE 17
004654  012777  100000  006670          MOV     #100000,@DHBAR          ;SET BAR BIT FOR LINE 17
004662  105777  006652          2$:     TSTB    @DHSCR                  ;WAIT FOR CHARACTER TO
004666  100375                          BPL     2$                      ;BE RECEIVED
004670  005201                          INC     R1                      ;UPDATE RECEIVED CHARACTER COUNT
004672  017704  006644                  MOV     @DHNRC,R4               ;READ CHARACTER
004676  020467  007444                  CMP     R4,TWRD17               ;IS CHARACTER CORRECT
004702  001406                          BEQ     3$
004704  016705  007436                  MOV     TWRD17,R5               ;(R5)=EXPECTED CHARACTER
004710  005077  006630                  CLR     @DHLPR                  ;SHUT OFF AUTO ECHO
004714                                  HLT     0                       ;CHARACTER ECHOED INCORRECTLY
004714  104000                          EMT     0
004716  000407                          BR      4$                      ;RESTART TEST
004720  005300                  3$:     DEC     R0                      ;IF 64 CHARACTERS HAVE NOT
004722  003357                          BGT     2$                      ;BEEN RECEIVED, CONTINUE
004724  100404                          BMI     4$
004726  042777  100000  006610          BIC     #100000,@DHLPR          ;SHUT OFF AUTO-ECHO
004734  000752                          BR      2$                      ;GET 1 MORE CHARACTER
004736  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000020                  LINE=LINE+1
        000000                  BITX=BITX+BITX
        000020                  KX=KX+1
```

```
                        XLINE=LINE
                        XBIT=BITX
                        K=KX
                        LINE=0
                        BITX=1
                        KX=0
                        .REPT   20
                        AUTO2   \LINE,\BITX,\KX
                        .NLIST
                        LINE=LINE+1
                        BITX=BITX+BITX
                        KX=KX+1
                        .LIST
                        .ENDR
004740                  AUTO2   \LINE,\BITX,\KX

                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 0
                        ;TRANSMIT 1 CHARACTER ON LINE 0 WITH AUTO ECHO ENABLED
                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                        ;CHARACTER LENGTH IS 8 BITS
                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

004740                  TS \XN,10,5$
004740  012767  000340  173030  T21:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
004746  012767  000010  006632          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
004754  012767  005174  006620          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                        .ENDC
        000022          XN=XN+1
004762  012777  004000  006550          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
004770  004767  006432                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
004774  012777  000000  006536          MOV     #0,@DHSCR               ;SELECT LINE 0 FOR TESTING
005002  012777  014310  006536          MOV     #TWRDO,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 0 IN AUTO ECHO MODE
005010  012777  177777  006532          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 0
005016  012777  133503  006520          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 0
005024  042767  000001  006610          BIC     #1,LINACT       ;CLEAR LINE ACTIVE BIT
005032  012777  177777  006512          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
005040  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
005042  017704  006474          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
005046  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
00505C  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
005052  000303                          SWAB    R3
005054  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
005060  010302                          MOV     R3,R2
005062  006302                          ASL     R2
005064  020327  000000                  CMP     R3,#0           ;IF LINE NUMBER IS 0
005070  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
005072  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 0, CHECK DATA
005076  001404                          BEQ     2$
005100  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
005104                                  HLT     1                       ;NON ECHOED DATA ERROR
005104  104001                          EMT     1
005106  000423                          BR      4$
005110  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
```

```
005114  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
005116  046267  014450  006516          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
005124  005767  006512          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
005130  001344                          BNE     1$                      ;EXIT
005132  012777  000000  006400          MOV     #0,@DHSCR               ;SELECT LINE 0
005140  042777  100000  006376          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 0
005146  105777  006406                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
005152  001333                          BNE     1$                      ;AND CHECK
005154  000407                          BR      5$
005156  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
005160  020467  007124                  CMP     R4,TWRD0                ;CHECK ECHOED DATA
005164  001757                          BEQ     3$
005166  016705  007116                  MOV     TWRD0,R5                        ;(R5)=EXPECTED ECHOED DATA
005172                                  HLT     2                       ;ECHOED DATA ERROR
005172  104002                          EMT     2
005174  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000001                  LINE=LINE+1
        000002                  BITX=BITX+BITX
        000001                  KX=KX+1
005176                          AUTO2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 1
                                ;TRANSMIT 1 CHARACTER ON LINE 1 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

005176                          TS \XN,10,5$
005176  012767  000340  172572  T22:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
005204  012767  000010  006374          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
005212  012767  005432  006362          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                        .ENDC
        000023          XN=XN+1
005220  012777  004000  006312          MOV     #6IT11,@DHSCR           ;MASTER CLEAR INTERFACE
005226  004767  006174                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
005232  012777  000001  006300          MOV     #1,@DHSCR               ;SELECT LINE 1 FOR TESTING
005240  012777  014312  006300          MOV     #TWRD1,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 1 IN AUTO ECHO MODE
005246  012777  177777  006274          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 1
005254  012777  133503  006262          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 1
005262  042767  000002  006352          BIC     #2,LINACT       ;CLEAR LINE ACTIVE BIT
005270  012777  177777  006254          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
005276  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
005300  017704  006236          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
005304  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
005306  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
005310  000303                          SWAB    R3
005312  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
005316  010302                          MOV     R3,R2
005320  006302                          ASL     R2
005322  020327  000001                  CMP     R3,#1           ;IF LINE NUMBER IS 1
005326  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
005330  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 1, CHECK DATA
005334  001404                          BEQ     2$
```

```
005336  016205  014246          MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
005342                          HLT     1                       ;NON ECHOED DATA ERROR
005342  104001                  EMT     1
005344  000423                  BR      4$
005346  105262  014246  2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
005352  001352                  BNE     1$                      ;CONTINUE IF NOT DONE
005354  046267  014450  006260  BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
005362  005767  006254  3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
005366  001344                  BNE     1$                      ;EXIT
005370  012777  000001  006142  MOV     #1,@DHSCR               ;SELECT LINE 1
005376  042777  100000  006140  BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 1
005404  105777  006150          TSTB    @DHSLR                  ;GET REST OF CHARACTERS
005410  001333                  BNE     1$                      ;AND CHECK
005412  000407                  BR      5$
005414  005200  4$:     INC     R0                              ;UPDATE ECHOED CHARACTER COUNT
005416  020467  006670          CMP     R4,TWRD1                ;CHECK ECHOED DATA
005422  001757                  BEQ     3$
005424  016705  006662          MOV     TWRD1,R5                        ;(R5)=EXPECTED ECHOED DATA
005430                          HLT     2                       ;ECHOED DATA ERROR
005430  104002                  EMT     2
005432  104400  5$:     SCOPE                                   ;CHECK FOR ITERATIONS, LOOP
        000002  LINE=LINE+1
        000004  BITX=BITX+BITX
        000002  KX=KX+1
005434          AUTO2   \LINE,\BITX,\KX

                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 2
                        ;TRANSMIT 1 CHARACTER ON LINE 2 WITH AUTO ECHO ENABLED
                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                        ;CHARACTER LENGTH IS 8 BITS
                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

005434          TS \XN,10,5$
005434  012767  000340  172334  T23:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
005442  012767  000010  006136          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
005450  012767  005670  006124          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                        .ENDC
        000024          XN=XN+1
005456  012777  004000  006054          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
005464  004767  005736                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
005470  012777  000002  006042          MOV     #2,@DHSCR               ;SELECT LINE 2 FOR TESTING
005476  012777  014314  006042          MOV     #TWRD2,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 2 IN AUTO ECHO MODE
005504  012777  177777  006036          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 2
005512  012777  133503  006024          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 2
005520  042767  000004  006114          BIC     #4,LINACT       ;CLEAR LINE ACTIVE BIT
005526  012777  177777  006016          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
005534  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
005536  017704  006000  1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
005542  100375                  BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
005544  010403                  MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
005546  000303                  SWAB    R3
005550  042703  177760          BIC     #177760,R3              ;CLEAR STATUS BITS
005554  010302                  MOV     R3,R2
```

```
005556  C06302                          ASL     R2
005560  020327  000002                  CMP     R3,#2           ;IF LINE NUMBER IS 2
005564  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
005566  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 2, CHECK DATA
005572  001404                          BEQ     2$
005574  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
005600                                  HLT     1                       ;NON ECHOED DATA ERROR
005600  104001                          EMT     1
005602  000423                          BR      4$
005604  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
005610  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
005612  046267  014450  006022          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
005620  005767  006016          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
005624  001344                          BNE     1$                      ;EXIT
005626  012777  000002  005704          MOV     #2,@DHSCR               ;SELECT LINE 2
005634  042777  100000  005702          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 2
005642  105777  005712                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
005646  001333                          BNE     1$                      ;AND CHECK
005650  000407                          BR      5$
005652  005200          4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
005654  020467  006434                  CMP     R4,TWRD2                ;CHECK ECHOED DATA
005660  001757                          BEQ     3$
005662  016705  006426                  MOV     TWRD2,R5                        ;(R5)=EXPECTED ECHOED DATA
005666                                  HLT     2                       ;ECHOED DATA ERROR
005666  104002                          EMT     2
005670  104400          5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000003          LINE=LINE+1
        000010          BITX=BITX+BITX
        000003          KX=KX+1
005672                  AUTO2   \LINE,\BITX,\KX

                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 3
                        ;TRANSMIT 1 CHARACTER ON LINE 3 WITH AUTO ECHO ENABLED
                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                        ;CHARACTER LENGTH IS 8 BITS
                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

005672                  TS \XN,10,5$
005672  012767  000340  172076  T24:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
005700  012767  000010  005700          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
005706  012767  006126  005666          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                        .ENDC
        000025          XN=XN+1
005714  012777  004000  005616          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
005722  004767  005500                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
005726  012777  000003  005604          MOV     #3,@DHSCR               ;SELECT LINE 3 FOR TESTING
005734  012777  014316  005604          MOV     #TWRD3,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 3 IN AUTO ECHO MODE
005742  012777  177777  005600          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 3
005750  012777  133503  005566          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 3
005756  042767  000010  005656          BIC     #10,LINACT      ;CLEAR LINE ACTIVE BIT
005764  012777  177777  005560          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
005772  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
005774  017704  005542          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
```

```
006000  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
006002  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
006004  000303                          SWAB    R3
006006  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
006012  010302                          MOV     R3,R2
006014  006302                          ASL     R2
006016  020327  000003                  CMP     R3,#3           ;IF LINE NUMBER IS 3
006022  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
006024  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 3, CHECK DATA
006030  001404                          BEQ     2$
006032  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
006036                                  HLT     1                       ;NON ECHOED DATA ERROR
006036  104001                          EMT     1
006040  000423                          BR      4$
006042  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
006046  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
006050  046267  014450  005564          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
006056  005767  005560          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
006062  001344                          BNE     1$                      ;EXIT
006064  012777  000003  005446          MOV     #3,@DHSCR               ;SELECT LINE 3
006072  042777  100000  005444          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 3
006100  105777  005454                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
006104  001333                          BNE     1$                      ;AND CHECK
006106  000407                          BR      5$
006110  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
006112  020467  006200                  CMP     R4,TWRD3                ;CHECK ECHOED DATA
006116  001757                          BEQ     3$
006120  016705  006172                  MOV     TWRD3,R5                        ;(R5)=EXPECTED ECHOED DATA
006124                                  HLT     2                       ;ECHOED DATA ERROR
006124  104002                          EMT     2
006126  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000004                  LINE=LINE+1
        000020                  BITX=BITX+BITX
        000004                  KX=KX+1
006130                          AUTO2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 4
                                ;TRANSMIT 1 CHARACTER ON LINE 4 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

006130                          TS \XN,10,5$
006130  012767  000340  171640  T25:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
006136  012767  000010  005442          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
006144  012767  006364  005430          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                        .ENDC
        000026          XN=XN+1
006152  012777  004000  005360          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
006160  004767  005242                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
006164  012777  000004  005346          MOV     #4,@DHSCR               ;SELECT LINE 4 FOR TESTING
006172  012777  014320  005346          MOV     #TWRD4,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 4 IN AUTO ECHO MODE
006200  012777  177777  005342          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 4
```

```
006206  012777  133503  005330          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 4
006214  042767  000020  005420          BIC     #20,LINACT      ;CLEAR LINE ACTIVE BIT
006222  012777  177777  005322          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
006230  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
006232  017704  005304          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
006236  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
006240  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
006242  000303                          SWAB    R3
006244  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
006250  010302                          MOV     R3,R2
006252  006302                          ASL     R2
006254  020327  000004                  CMP     R3,#4           ;IF LINE NUMBER IS 4
006260  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
006262  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 4, CHECK DATA
006266  001404                          BEQ     2$
006270  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
006274                                  HLT     1                       ;NON ECHOED DATA ERROR
006274  104001                          EMT     1
006276  000423                          BR      4$
006300  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
006304  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
006306  046267  014450  005326          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
006314  005767  005322          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
006320  001344                          BNE     1$                      ;EXIT
006322  012777  000004  005210          MOV     #4,@DHSCR               ;SELECT LINE 4
006330  042777  100000  005206          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 4
006336  105777  005216                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
006342  001333                          BNE     1$                      ;AND CHECK
006344  000407                          BR      5$
006346  005200          4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
006350  020467  005744                  CMP     R4,TWRD4                ;CHECK ECHOED DATA
006354  001757                          BEQ     3$
006356  016705  005736                  MOV     TWRD4,R5                        ;(R5)=EXPECTED ECHOED DATA
006362                                  HLT     2                       ;ECHOED DATA ERROR
006362  104002                          EMT     2
006364  104400          5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000005          LINE=LINE+1
        000040          BITX=BITX+BITX
        000005          KX=KX+1
006366                  AUTO2   \LINE,\BITX,\KX

                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 5
                        ;TRANSMIT 1 CHARACTER ON LINE 5 WITH AUTO ECHO ENABLED
                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                        ;CHARACTER LENGTH IS 8 BITS
                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

006366                  TS \XN,10,5$
006366  012767  000340  171402  T26:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
006374  012767  000010  005204          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
006402  012767  006622  005172          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                        .ENDC
        000027          XN=XN+1
006410  012777  004000  005122          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
006416  004767  005004                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
```

```
                                                        ;400 (OCTAL) CHARACTERS
006422  012777  000005  005110          MOV     #5,@DHSCR               ;SELECT LINE 5 FOR TESTING
006430  012777  014322  005110          MOV     #TWRD5,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                        ;ON LINE 5 IN AUTO ECHO MODE
006436  012777  177777  005104          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 5
006444  012777  133503  005072          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 5
006452  042767  000040  005162          BIC     #40,LINACT      ;CLEAR LINE ACTIVE BIT
006460  012777  177777  005064          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
006466  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
006470  017704  005046          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
006474  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
006476  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
006500  000303                          SWAB    R3
006502  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
006506  010302                          MOV     R3,R2
006510  006302                          ASL     R2
006512  020327  000005                  CMP     R3,#5           ;IF LINE NUMBER IS 5
006516  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
006520  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 5, CHECK DATA
006524  001404                          BEQ     2$
006526  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
006532                                  HLT     1                       ;NON ECHOED DATA ERROR
006532  104001                          EMT     1
006534  000423                          BR      4$
006536  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
006542  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
006544  046267  014450  005070          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
006552  005767  005064          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
006556  001344                          BNE     1$                      ;EXIT
006560  012777  000005  004752          MOV     #5,@DHSCR               ;SELECT LINE 5
006566  042777  100000  004750          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 5
006574  105777  004760                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
006600  001333                          BNE     1$                      ;AND CHECK
006602  000407                          BR      5$
006604  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
006606  020467  005510                  CMP     R4,TWRD5                ;CHECK ECHOED DATA
006612  001757                          BEQ     3$
006614  016705  005502                  MOV     TWRD5,R5                        ;(R5)=EXPECTED ECHOED DATA
006620                                  HLT     2                       ;ECHOED DATA ERROR
006620  104002                          EMT     2
006622  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000006                  LINE=LINE+1
        000100                  BITX=BITX+BITX
        000006                  KX=KX+1
006624                          AUTO2   \LINE,\BITX,\KX

                                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 6
                                        ;TRANSMIT 1 CHARACTER ON LINE 6 WITH AUTO ECHO ENABLED
                                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                        ;CHARACTER LENGTH IS 8 BITS
                                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

006624                          TS \XN,10,5$
006624  012767  000340  171144  T27:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
006632  012767  000010  004746          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
006640  012767  007060  004734          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
```

```
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
        000030                  XN=XN+1
006646  012777  004000  004664         MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
006654  004767  004546                 JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
006660  012777  000006  004652         MOV     #6,@DHSCR               ;SELECT LINE 6 FOR TESTING
006666  012777  014324  004652         MOV     #TWRD6,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 6 IN AUTO ECHO MODE
006674  012777  177777  004646         MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 6
006702  012777  133503  004634         MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 6
006710  042767  000100  004724         BIC     #100,LINACT     ;CLEAR LINE ACTIVE BIT
006716  012777  177777  004626         MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
006724  005000                         CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
006726  017704  004610         1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
006732  100375                         BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
006734  010403                         MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
006736  000303                         SWAB    R3
006740  042703  177760                 BIC     #177760,R3              ;CLEAR STATUS BITS
006744  010302                         MOV     R3,R2
006746  006302                         ASL     R2
006750  020327  000006                 CMP     R3,#6           ;IF LINE NUMBER IS 6
006754  001432                         BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
006756  026204  014246                 CMP     RBUF(R2),R4             ;IF NOT LINE 6, CHECK DATA
006762  001404                         BEQ     2$
006764  016205  014246                 MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
006770                                 HLT     1                       ;NON ECHOED DATA ERROR
006770  104001                         EMT     1
006772  000423                         BR      4$
006774  105262  014246         2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
007000  001352                         BNE     1$                      ;CONTINUE IF NOT DONE
007002  046267  014450  004632         BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
007010  005767  004626         3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
007014  001344                         BNE     1$                      ;EXIT
007016  012777  000006  004514         MOV     #6,@DHSCR               ;SELECT LINE 6
007024  042777  100000  004512         BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 6
007032  105777  004522                 TSTB    @DHSLR                  ;GET REST OF CHARACTERS
007036  001333                         BNE     1$                      ;AND CHECK
007040  000407                         BR      5$
007042  005200         4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
007044  020467  005254                 CMP     R4,TWRD6                ;CHECK ECHOED DATA
007050  001757                         BEQ     3$
007052  016705  005246                 MOV     TWRD6,R5                        ;(R5)=EXPECTED ECHOED DATA
007056                                 HLT     2                       ;ECHOED DATA ERROR
007056  104002                         EMT     2
007060  104400         5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
                        LINE=LINE+1
                        BITX=BITX+BITX
                        KX=KX+1
007062                  AUTO2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 7
                                ;TRANSMIT 1 CHARACTER ON LINE 7 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES
```

```
                                        TS \XN,10 ;
007062                                  T30:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
007062  012767  000340  170706                  MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
007070  012767  000010  004510                  MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
007076  012767  007316  004476                  .IF NB  <>
                                                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                                .ENDC
        000031                                  XN=XN+1
007104  012777  004000  004426                  MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
007112  004767  004310                          JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                                ;400 (OCTAL) CHARACTERS
007116  012777  000007  004414                  MOV     #7,@DHSCR               ;SELECT LINE 7 FOR TESTING
007124  012777  014326  004414                  MOV     #TWRD7,@DHBA    ;CHARACTER TO BE TRANSMITTED
                                                                                ;ON LINE 7 IN AUTO ECHO MODE
007132  012777  177777  004410                  MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 7
007140  012777  133503  004376                  MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 7
007146  042767  000200  004466                  BIC     #200,LINACT     ;CLEAR LINE ACTIVE BIT
007154  012777  177777  004370                  MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
007162  005000                                  CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
007164  017704  004352          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
007170  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
007172  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
007174  000303                          SWAB    R3
007176  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
007202  010302                          MOV     R3,R2
007204  006302                          ASL     R2
007206  020327  000007                  CMP     R3,#7           ;IF LINE NUMBER IS 7
007212  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
007214  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 7, CHECK DATA
007220  001404                          BEQ     2$
007222  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
007226                                  HLT     1                       ;NON ECHOED DATA ERROR
007226  104001                          EMT     1
007230  000423                          BR      4$
007232  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
007236  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
007240  046267  014450  004374                  BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
007246  005767  004370          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
007252  001344                          BNE     1$                      ;EXIT
007254  012777  000007  004256                  MOV     #7,@DHSCR               ;SELECT LINE 7
007262  042777  100000  004254                  BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 7
007270  105777  004264                          TSTB    @DHSLR                  ;GET REST OF CHARACTERS
007274  001333                          BNE     1$                      ;AND CHECK
007276  000407                          BR      5$
007300  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
007302  020467  005020                  CMP     R4,TWRD7                ;CHECK ECHOED DATA
007306  001757                          BEQ     3$
007310  016705  005012                  MOV     TWRD7,R5                        ;(R5)=EXPECTED ECHOED DATA
007314                                  HLT     2                       ;ECHOED DATA ERROR
007314  104002                          EMT     2
007316  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000010                  LINE=LINE+1
        000400                  BITX=BITX+BITX
        000010                  KX=KX+1
007320                          AUTO2   \LINE,\BITX,\KX

                                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 10
```

```
                                        ;TRANSMIT 1 CHARACTER ON LINE 10 WITH AUTO ECHO ENABLED
                                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                        ;CHARACTER LENGTH IS 8 BITS
                                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

 007320                                 TS \XN,10,5$
 007320  012767  000340  170450  T31:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
 007326  012767  000010  004252        MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
 007334  012767  007554  004240        MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                        .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                        .ENDC
         000032                         XN=XN+1
 007342  012777  004000  004170        MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 007350  004767  004052                JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                       ;400 (OCTAL) CHARACTERS
 007354  012777  000010  004156        MOV     #10,@DHSCR              ;SELECT LINE 10 FOR TESTING
 007362  012777  014330  004156        MOV     #TWRD10,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                       ;ON LINE 10 IN AUTO ECHO MODE
 007370  012777  177777  004152        MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 10
 007376  012777  133503  004140        MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 10
 007404  042767  000400  004230        BIC     #400,LINACT     ;CLEAR LINE ACTIVE BIT
 007412  012777  177777  004132        MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
 007420  005000                        CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
 007422  017704  004114        1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
 007426  100375                        BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
 007430  010403                        MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
 007432  000303                        SWAB    R3
 007434  042703  177760                BIC     #177760,R3              ;CLEAR STATUS BITS
 007440  010302                        MOV     R3,R2
 007442  006302                        ASL     R2
 007444  020327  000010                CMP     R3,#10          ;IF LINE NUMBER IS 10
 007450  001432                        BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
 007452  026204  014246                CMP     RBUF(R2),R4             ;IF NOT LINE 10, CHECK DATA
 007456  001404                        BEQ     2$
 007460  016205  014246                MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
 007464                                HLT     1                       ;NON ECHOED DATA ERROR
 007464  104001                        EMT     1
 007466  000423                        BR      4$
 007470  105262  014246        2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
 007474  001352                        BNE     1$                      ;CONTINUE IF NOT DONE
 007476  046267  014450  004136        BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
 007504  005767  004132        3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
 007510  001344                        BNE     1$                      ;EXIT
 007512  012777  000010  004020        MOV     #10,@DHSCR              ;SELECT LINE 10
 007520  042777  100000  004016        BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 10
 007526  105777  004026                TSTB    @DHSLR                  ;GET REST OF CHARACTERS
 007532  001333                        BNE     1$                      ;AND CHECK
 007534  000407                        BR      5$
 007536  005200        4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
 007540  020467  004564                CMP     R4,TWRD10               ;CHECK ECHOED DATA
 007544  001757                        BEQ     3$
 007546  016705  004556                MOV     TWRD10,R5                       ;(R5)=EXPECTED ECHOED DATA
 007552                                HLT     2                       ;ECHOED DATA ERROR
 007552  104002                        EMT     2
 007554  104400        5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
         000011                 LINE=LINE+1
```

```
                001000                  BITX=BITX+BITX
                000011                  KX=KX+1
007556                                  AUTO2   \LINE,\BITX,\KX

                                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 11
                                        ;TRANSMIT 1 CHARACTER ON LINE 11 WITH AUTO ECHO ENABLED
                                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                        ;CHARACTER LENGTH IS 8 BITS
                                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

007556                                  TS \XN,10,5$
007556  012767  000340  170212  T32:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
007564  012767  000010  004014          MOV     #10.,ICOUNT             ;SET UP FOR 10 ITERATIONS
007572  012767  010012  004002          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
                000033                  XN=XN+1
007600  012777  004000  003732          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
007606  004767  003614                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
007612  012777  000011  003720          MOV     #11,@DHSCR              ;SELECT LINE 11 FOR TESTING
007620  012777  014332  003720          MOV     #TWRD11,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 11 IN AUTO ECHO MODE
007626  012777  177777  003714          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 11
007634  012777  133503  003702          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 11
007642  042767  001000  003772          BIC     #1000,LINACT    ;CLEAR LINE ACTIVE BIT
007650  012777  177777  003674          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
007656  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
007660  017704  003656          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
007664  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
007666  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
007670  000303                          SWAB    R3
007672  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
007676  010302                          MOV     R3,R2
007700  006302                          ASL     R2
007702  020327  000011                  CMP     R3,#11          ;IF LINE NUMBER IS 11
007706  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
007710  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 11, CHECK DATA
007714  001404                          BEQ     2$
007716  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
007722                                  HLT     1                       ;NON ECHOED DATA ERROR
007722  104001                          EMT     1
007724  000423                          BR      4$
007726  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
007732  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
007734  046267  014450  003700          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
007742  005767  003674          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
007746  001344                          BNE     1$                      ;EXIT
007750  012777  000011  003562          MOV     #11,@DHSCR              ;SELECT LINE 11
007756  042777  100000  003560          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 11
007764  105777  003570                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
007770  001333                          BNE     1$                      ;AND CHECK
007772  000407                          BR      5$
007774  005200          4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
007776  020467  004330                  CMP     R4,TWRD11               ;CHECK ECHOED DATA
010002  001757                          BEQ     3$
```

```
010004  016705  004322          MOV     TWRD11,R5                       ;(R5)=EXPECTED ECHOED DATA
010010                          HLT     2                       ;ECHOED DATA ERROR
010010  104002                  EMT     2
010012  104400          5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000012          LINE=LINE+1
        002000          BITX=BITX+BITX
        000012          KX=KX+1
010014                  AUTO2   \LINE,\BITX,\KX

                        ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 12
                        ;TRANSMIT 1 CHARACTER ON LINE 12 WITH AUTO ECHO ENABLED
                        ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                        ;CHARACTER LENGTH IS 8 BITS
                        ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

010014                  TS \XN,10,5$
010014  012767  000340  167754  T33:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
010022  012757  000010  003556          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
010030  012767  010250  003544          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                        MOV     #.,FREEZ1               ;SET UP TO LOOP WITH DATA       ; 3
                        .ENDC
        000034          XN=XN+1
010036  012777  004000  003474          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
010044  004767  003356                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
010050  012777  000012  003462          MOV     #12,@DHSCR              ;SELECT LINE 12 FOR TESTING
010056  012777  014334  003462          MOV     #TWRD12,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 12 IN AUTO ECHO MODE
010064  012777  177777  003456          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 12
010072  012777  133503  003444          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 12
010100  042767  002000  003534          BIC     #2000,LINACT    ;CLEAR LINE ACTIVE BIT
010106  012777  177777  003436          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
010114  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
010116  017704  003420          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
010122  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
010124  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
010126  000303                          SWAB    R3
010130  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
010134  010302                          MOV     R3,R2
010136  006302                          ASL     R2
010140  020327  000012                  CMP     R3,#12          ;IF LINE NUMBER IS 12
010144  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
010146  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 12, CHECK DATA
010152  001404                          BEQ     2$
010154  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
010160                          HLT     1                       ;NON ECHOED DATA ERROR
010160  104001                  EMT     1
010162  000423                  BR      4$
010164  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
010170  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
010172  046267  014450  003442          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
010200  005767  003436          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
010204  001344                          BNE     1$                      ;EXIT
010206  012777  000012  003324          MOV     #12,@DHSCR              ;SELECT LINE 12
010214  042777  100000  003322          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 12
010222  105777  003332                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
```

```
010226  001333                          BNE     1$                      ;AND CHECK
010230  000407                          BR      5$
010232  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
010234  020467  004074                  CMP     R4,TWRD12               ;CHECK ECHOED DATA
010240  001757                          BEQ     3$
010242  016705  004066                  MOV     TWRD12,R5                       ;(R5)=EXPECTED ECHOED DATA
010246                                  HLT     2                       ;ECHOED DATA ERROR
010246  104002                          EMT     2
010250  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000013                  LINE=LINE+1
        004000                  BITX=BITX+BITX
        000013                  KX=KX+1
010252                          AUTC2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 13
                                ;TRANSMIT 1 CHARACTER ON LINE 13 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

010252                          TS \XN,10,5$
010252  012767  000340  167516  T34:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
010260  012767  000010  003320          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
010266  012767  010506  003306          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000035                  XN=XN+1
010274  012777  004000  003236          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
010302  004767  003120                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
010306  012777  000013  003224          MOV     #13,@DHSCR              ;SELECT LINE 13 FOR TESTING
010314  012777  014336  003224          MOV     #TWRD13,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 13 IN AUTO ECHO MODE
010322  012777  177777  003220          MOV     #-1,@DHSC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 13
010330  012777  133503  003206          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 13
010336  042767  004000  003276          BIC     #4000,LINACT    ;CLEAR LINE ACTIVE BIT
010344  012777  177777  003200          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
010352  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
010354  017704  003162          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
010360  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
010362  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
010364  000303                          SWAB    R3
010366  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
010372  010302                          MOV     R3,R2
010374  006302                          ASL     R2
010376  020327  000013                  CMP     R3,#13          ;IF LINE NUMBER IS 13
010402  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
010404  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 13, CHECK DATA
010410  001404                          BEQ     2$
010412  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
010416                                  HLT     1                       ;NON ECHOED DATA ERROR
010416  104001                          EMT     1
010420  000423                          BR      4$
010422  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
010426  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
010430  046267  014450  003204          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
```

```
010436  005767  003200          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
010442  001344                          BNE     1$                      ;EXIT
010444  012777  000013  003066          MOV     #13,@DHSCR              ;SELECT LINE 13
010452  042777  100000  003064          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 13
010460  105777  003074                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
010464  001333                          BNE     1$                      ;AND CHECK
010466  000407                          BR      5$
010470  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
010472  020467  003640                  CMP     R4,TWRD13               ;CHECK ECHOED DATA
010476  001757                          BEQ     3$
010500  016705  003632                  MOV     TWRD13,R5                       ;(R5)=EXPECTED ECHOED DATA
010504                                  HLT     2                       ;ECHOED DATA ERROR
010504  104002                          EMT     2
010506  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000014                  LINE=LINE+1
        010000                  BITX=BITX+BITX
        000014                  KX=KX+1
010510                          AUTO2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 14
                                ;TRANSMIT 1 CHARACTER ON LINE 14 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

010510                          TS \XN,10,5$
010510  012767  000340  167260  T35:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
010516  012767  000010  003062          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
010524  012767  010744  003050          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000036                  XN=XN+1
010532  012777  004000  003000          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
010540  004767  002662                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
010544  012777  000014  002766          MOV     #14,@DHSCR              ;SELECT LINE 14 FOR TESTING
010552  012777  014340  002766          MOV     #TWRD14,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 14 IN AUTO ECHO MODE
010560  012777  177777  002762          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 14
010566  012777  133503  002750          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 14
010574  042767  010000  003040          BIC     #10000,LINACT   ;CLEAR LINE ACTIVE BIT
010602  012777  177777  002742          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
010610  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
010612  017704  002724          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
010616  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
010620  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
010622  000303                          SWAB    R3
010624  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
010630  010302                          MOV     R3,R2
010632  006302                          ASL     R2
010634  020327  000014                  CMP     R3,#14          ;IF LINE NUMBER IS 14
010640  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
010642  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 14, CHECK DATA
010646  001404                          BEQ     2$
010650  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
010654                                  HLT     1                       ;NON ECHOED DATA ERROR
```

```
010654  104001                          EMT     1
010656  000423                          BR      4$
010660  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
010664  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
010666  046267  014450  002746          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
010674  005767  002742          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
010700  001344                          BNE     1$                      ;EXIT
010702  012777  000014  002630          MOV     #14,@DHSCR              ;SELECT LINE 14
010710  042777  100000  002626          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 14
010716  105777  002636                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
010722  001333                          BNE     1$                      ;AND CHECK
010724  000407                          BR      5$
010726  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
010730  020467  003404                  CMP     R4,TWRD14               ;CHECK ECHOED DATA
010734  001757                          BEQ     3$
010736  016705  003376                  MOV     TWRD14,R5                       ;(R5)=EXPECTED ECHOED DATA
010742                                  HLT     2                       ;ECHOED DATA ERROR
010742  104002                          EMT     2
010744  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS  LOOP
        000015                  LINE=LINE+1
        020000                  BITX=BITX+BITX
        000015                  KX=KX+1
010746                          AUTO2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 15
                                ;TRANSMIT 1 CHARACTER ON LINE 15 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

010746                          TS \XN,10,5$
010746  012767  000340  167022  T36:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
010754  012767  000010  002624          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
010762  012767  011202  002612          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA         ; 3
                                .ENDC
        000037                  XN=XN+1
010770  012777  004000  002542          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
010776  004767  002424                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
011002  012777  000015  002530          MOV     #15,@DHSCR              ;SELECT LINE 15 FOR TESTING
011010  012777  014342  002530          MOV     #TWRD15,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 15 IN AUTO ECHO MODE
011016  012777  177777  002524          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 15
011024  012777  133503  002512          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 15
011032  042767  020000  002602          BIC     #20000,LINACT   ;CLEAR LINE ACTIVE BIT
011040  012777  177777  002504          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
011046  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
011050  017704  002466          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
011054  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
011056  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
011060  000303                          SWAB    R3
011062  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
011066  010302                          MOV     R3,R2
011070  006302                          ASL     R2
011072  020327  000015                  CMP     R3,#15          ;IF LINE NUMBER IS 15
```

```
011076  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
011100  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 15, CHECK DATA
011104  001404                          BEQ     2$
011106  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
011112                                  HLT     1                       ;NON ECHOED DATA ERROR
011112  104001                          EMT     1
011114  000423                          BR      4$
011116  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
011122  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
011124  046267  014450  002510          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
011132  005767  002504          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
011136  001344                          BNE     1$                      ;EXIT
011140  012777  000015  002372          MOV     #15,@DHSCR              ;SELECT LINE 15
011146  042777  100000  002370          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 15
011154  105777  002400                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
011160  001323                          BNE     1$                      ;AND CHECK
011162  000407                          BR      5$
011164  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
011166  020467  003150                  CMP     R4,TWRD15               ;CHECK ECHOED DATA
011172  001757                          BEQ     3$
011174  016705  003142                  MOV     TWRD15,R5                       ;(R5)=EXPECTED ECHOED DATA
011200                                  HLT     2                       ;ECHOED DATA ERROR
011200  104002                          EMT     2
011202  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000016                  LINE=LINE+1
        040000                  BITX=BITX+BITX
        000016                  KX=KX+1
011204                          AUTO2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 16
                                ,TRANSMIT 1 CHARACTER ON LINE 16 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

011204                          TS \YN,10,5$
011204  012767  000340  166564  T37:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
011212  012767  000010  002366          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
011220  012767  011440  002354          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
        000040                  XN=XN+1
011226  012777  004000  002304          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
011234  004767  002166                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
011240  012777  000016  002272          MOV     #16,@DHSCR              ;SELECT LINE 16 FOR TESTING
011246  012777  014344  002272          MOV     #TWRD16,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 16 IN AUTO ECHO MODE
011254  012777  177777  002266          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 16
011262  012777  133503  002254          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 16
011270  042767  040000  002344          BIC     #40000,LINACT   ;CLEAR LINE ACTIVE BIT
011276  012777  177777  002246          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
011304  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
011306  017704  002230          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
011312  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
011314  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
```

```
011316  000303                          SWAB    R3
011320  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
011324  010302                          MOV     R3,R2
011326  006302                          ASL     R2
011330  020327  000016                  CMP     R3,#16          ;IF LINE NUMBER IS 16
011334  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
011336  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 16, CHECK DATA
011342  001404                          BEQ     2$
011344  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
011350                                  HLT     1                       ;NON ECHOED DATA ERROR
011350  104001                          EMT     1
011352  000423                          BR      4$
011354  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
011360  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
011362  046267  014450  002252          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
011370  005767  002246          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
011374  001344                          BNE     1$                      ;EXIT
011376  012777  000016  002134          MOV     #16,@DHSCR              ;SELECT LINE 16
011404  042777  100000  002132          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 16
011412  105777  002142                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
011416  001333                          BNE     1$                      ;AND CHECK
011420  000407                          BR      5$
011422  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
011424  020467  002714                  CMP     R4,TWRD16               ;CHECK ECHOED DATA
011430  001757                          BEQ     3$
011432  016705  002706                  MOV     TWRD16,R5                       ;(R5)=EXPECTED ECHOED DATA
011436                                  HLT     2                       ;ECHOED DATA ERROR
011436  104002                          EMT     2
011440  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000017                  LINE=LINE+1
        100000                  BITX=BITX+BITX
        000017                  KX=KX+1
011442                          AUTO2   \LINE,\BITX,\KX

                                ;TRANSMIT A BINARY COUNT PATTERN ON ALL LINES EXCEPT LINE 17
                                ;TRANSMIT 1 CHARACTER ON LINE 17 WITH AUTO ECHO ENABLED
                                ;TRANSMISSION SPEED FOR ALL LINES IS 9600 BAUD
                                ;CHARACTER LENGTH IS 8 BITS
                                ;VERIFY THAT CORRECT DATA IS RECEIVED ON ALL LINES

011442                          TS \XN-10,5$
011442  012767  000340  166326  T40:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
011450  012767  000010  002130          MOV     #10,ICOUNT              ;SET UP FOR 10 ITERATIONS
011456  012767  011676  002116          MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
        000041                  XN=XN+1
011464  012777  004000  002046          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
011472  004767  001730                  JSR     PC,SETALL               ;SET UP ALL LINES TO TRANSMIT
                                                                        ;400 (OCTAL) CHARACTERS
011476  012777  000017  002034          MOV     #17,@DHSCR              ;SELECT LINE 17 FOR TESTING
011504  012777  014346  002034          MOV     #TWRD17,@DHBA   ;CHARACTER TO BE TRANSMITTED
                                                                        ;ON LINE 17 IN AUTO ECHO MODE
011512  012777  177777  002030          MOV     #-1,@DHBC               ;TRANSMIT ONLY 1 CHARACTER ON LINE 17
011520  012777  133503  002016          MOV     #133503,@DHLPR          ;SET AUTO ECHO FOR LINE 17
011526  042767  100000  002106          BIC     #100000,LINACT  ;CLEAR LINE ACTIVE BIT
```

```
011534  012777  177777  002010          MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
011542  005000                          CLR     R0                      ;KEEP COUNT OF NUMBER OF RECEIVED CHARACTERS
011544  017704  001772          1$:     MOV     @DHNRC,R4               ;GET A CHARACTER FROM SILO
011550  100375                          BPL     1$                      ;IF NOT VALID DATA, TRY AGAIN
011552  010403                          MOV     R4,R3                   ;EXTRACT LINE NUMBER FORM CHARACTER
011554  000303                          SWAB    R3
011556  042703  177760                  BIC     #177760,R3              ;CLEAR STATUS BITS
011562  010302                          MOV     R3,R2
011564  006302                          ASL     R2
011566  020327  000017                  CMP     R3,#17          ;IF LINE NUMBER IS 17
011572  001432                          BEQ     4$                      ;CHECK FOR CORRECT ECHOED CHARACTER
011574  026204  014246                  CMP     RBUF(R2),R4             ;IF NOT LINE 17, CHECK DATA
011600  001404                          BEQ     2$
011602  016205  014246                  MOV     RBUF(R2),R5             ;(R5)=EXPECTED NON ECHOED DATA
011606                                  HLT     1                       ;NON ECHOED DATA ERROR
011606  104001                          EMT     1
011610  000423                          BR      4$
011612  105262  014246          2$:     INCB    RBUF(R2)                ;UPDATE EXPECTED RECEIVED DATA
011616  001352                          BNE     1$                      ;CONTINUE IF NOT DONE
011620  046267  014450  002014          BIC     LINBIT(R2),LINACT       ;CLEAR ACTIVE BIT
011626  005767  002010          3$:     TST     LINACT                  ;IF ALL LINES ARE DONE
011632  001344                          BNE     1$                      ;EXIT
011634  012777  000017  001676          MOV     #17,@DHSCR              ;SELECT LINE 17
011642  042777  100000  001674          BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO FOR LINE 17
011650  105777  001704                  TSTB    @DHSLR                  ;GET REST OF CHARACTERS
011654  001333                          BNE     1$                      ;AND CHECK
011656  000407                          BR      5$
011660  005200                  4$:     INC     R0                      ;UPDATE ECHOED CHARACTER COUNT
011662  020467  002460                  CMP     R4,TWRD17               ;CHECK ECHOED DATA
011666  001757                          BEQ     3$
011670  016705  002452                  MOV     TWRD17,R5
                                                        ;(R5)=EXPECTED ECHOED DATA
011674                                  HLT     2                       ;ECHOED DATA ERROR
011674  104002                          EMT     2
011676  104400                  5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
        000020                  LINE=LINE+1
        000000                  BITX=BITX+BITX
        000020                  KX=KX+1
```

```
                                        ;TRANSMIT 1 CHARACTER ON EACH LINE WITH AUTO ECHO ENABLED
                                        ;RECEIVE 64 CHARACTERS ON EACH LINE

 5 011700                       TS \XN,100,5$
   011700  012767  000340  166070 T41:    MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
   011706  012767  000100  001672        MOV     #100,ICOUNT             ;SET UP FOR 100 ITERATIONS
   011714  012767  012130  001660        MOV     #5$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                        MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA       ; 3
                                .ENDC
           000042                       XN=XN+1
 6 011722  012777  004000  001610        MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 7 011730  012700  000020                MOV     #20,R0                  ;SET UP PARAMETERS FOR 16 LINES
 8 011734  012701  014310                MOV     #TWRDO,R1               ;CHARACTER TO BE TRANSMITTED
 9 011740  012702  014350                MOV     #RCNTO,R2               ;RECEIVED CHARACTER COUNT
10 011744  012703  014410                MOV     #RDCTO,R3               ;EXPECTED NUMBER OF CHARACTERS
11 011750  010177  001512       1$:      MOV     R1,@DHBA                ;LOAD BUS ADDRESS
12 011754  012777  177777  001566        MOV     #-1,@DHBC               ;LOAD BYTE COUNT
13 011762  012777  131403  001554        MOV     #131403,@DHLPR          ;SET AUTO ECHO
14 011770  005022                        CLR     (R2)+                   ;CLEAR RECEIVED CHARACTER COUNT
15 011772  012723  000100                MOV     #100,(R3)+              ;=NUMBER OF CHARACTERS TO BE RECEIVED
16 011776  062701  000002                ADD     #2,R1                   ;ADVANCE POINTER
17 012002  005300                        DEC     R0                      ;CONTINUE IF NOT DONE
18 012004  001361                        BNE     1$
19 012006  012767  177777  001630        MOV     #-1,AEACT               ;INDICATE AUTO ECHO ACTIVE
20                                                                      ;FOR ALL LINES
21 012014  012777  177777  001530        MOV     #-1,@DHBAR              ;SET BAR BITS FOR ALL LINES
22 012022  105777  001512       2$:      TSTB    @DHSCR                  ;WAIT FOR A CHARACTER
23 012026  100375                        BPL     2$
24 012030  017704  001506                MOV     @DHNRC,R4               ;GET CHARACTER
25 012034  010403                        MOV     R4,R3
26 012036  000303                        SWAB    R3
27 012040  042703  177760                BIC     #177760,R3              ;EXTRACT LINE NUMBER
28 012044  010302                        MOV     R3,R2
29 012046  006302                        ASL     R2
30 012050  005262  014350                INC     RCNTO(R2)               ;UPDATE RECEIVED COUNT FOR LINE
31 012054  020462  014310                CMP     R4,TWRDO(R2)            ;CHECK EXPECTED AND RECEIVED DATA
32 012060  001404                        BEQ     3$
33 012062  016205  014310                MOV     TWRDO(R2),R5            ;(R5)=EXPECTED ECHOED DATA
34 012066                                HLT     2                       ;AUTO ECHO ERROR
   012066  104002                        EMT     2
35 012070  000417                        BR      5$
36 012072  005362  014410       3$:      DEC     RDCTO(R2)               ;UPDATE RECEIVED EXPECTED COUNT
37 012076  003351                        BGT     2$                      ;CONTINUE IF NOT 0
38 012100  100413                        BMI     5$                      ;EXIT IOF NEGATIVE
39 012102  010377  001432                MOV     R3,@DHSCR               ;SELECT LINE THAT FINISHED
40 012106  042777  100000  001430        BIC     #100000,@DHLPR          ;CLEAR AUTO ECHO
41 012114  046267  014450  001522        BIC     LINBIT(R2),AEACT        ;CLEAR AUTO ECHO ACTIVE
42 012122  005767  001516                TST     AEACT                   ;ALL LINES DONE
43 012126  001335                        BNE     2$                      ;IF NOT, CONTINUE
44 012130  104400                5$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
```

```
      1
      2 012132                          .EOP    ↑/BEGIN/

                                                ;END OF PASS
                                                ;TYPE NAME OF TEST
                                                ;UPDATE PASS COUNT
                                                ;CHECK FOR EXIT TO ACT-11
                                                ;RESTART TEST

   012132  104401                       EOP:    TYPE                            ;TYPE NAME OF TEST
   012134  015064                               MEPASS
   012136  005067  001474                       CLR     LAST                    ;CLEAR LAST ERROR PC
   012142  005067  001424                       CLR     ERRFLG                  ;CLEAR ERROR FLAG
   012146  005267  001422                       INC     PASCNT                  ;UPDATE PASS COUNT
   012152  005767  166624                       TST     LIGHTS                  ; ARE WE USING LIGHTS?          ; 4
   012156  001005                               BNE     2$                      ; BRANCH IF WE ARE              ; 6
   012160  104401                               TYPE                            ; TYPE PASCOUNT MESSAGE         ; 5
   012162  015077                               PASTXT                                                          ; 5
   012164  104402                               OCTASC                          ; PRINT PASSCOUNT               ; 4
   012166  012224                               PASARG                                                                  ;4 ; 6
   012170  000403                               BR      3$                      ; CONTINUE                      ; 4
   012172                                2$:                                                                     ; 4
   012172  016767  001376  166602               MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
   012200                                3$:                                                                     ; 4
   012200  013701  000042                       MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
   012204  001405                               BEQ     RESTRT                  ;IF NOT, CONTINUE TESTING
   012206  000005                               RESET
   012210  004711                       LOGICAL:        JSR     PC,(R1)
   012212  000240                               NOP
   012214  000240                               NOP
   012216  000240                               NOP
   012220  000167  167062               RESTRT: JMP     BEGIN
   012224  000001                       PASARG: .WORD   1                       ; PARAMETERS TO PRINT PASSCOUNT  ; 5
   012226     006     002                       .BYTE   6,2                                                     ; 5
   012230  013574                               .WORD   PASCNT                                                  ; 5
      3 012232                          .SCOPE

                                                ;CHECK FOR LOOP ON CURRENT TEST                                 ; 3
                                                ;CHECK FOR ITERATION SUPPRESSION

   012232  032777  002000  166540       SCOPER: BIT     #SW10,@SWR                                              ; 4
   012240  001030                               BNE     4$
   012242  032777  040000  166530       1$:     BIT     #SW14,@SWR                                              ; 4
   012250  001021                               BNE     3$
   012252  032777  004000  166520               BIT     #SW11,@SWR                                              ; 4
   012260  001006                               BNE     2$
   012262  005267  001322                       INC     LPCNT
   012266  026767  001316  001312               CMP     LPCNT,ICOUNT
   012274  001007                               BNE     3$
   012276  005067  001306               2$:     CLR     LPCNT
   012302  005067  001264                       CLR     ERRFLG
   012306  011667  001266                       MOV     (SP),RETRN
   012312  000002                               RTI
   012314  016716  001260               3$:     MOV     RETRN,(SP)
   012320  000002                               RTI
   012322  005767  001244               4$:     TST     ERRFLG
   012326  001745                               BEQ     1$
```

```
  012330  000762                          BR      2$
4 012332                          .SCOP1

                                  ;CHECK FOR FREEZE ON CURRENT DATA

  012332  032777  001000  166440  SCOP1R: BIT     @SW09,@SWR                      ; 4
  012340  001402                          BEQ     1$
  012342  016716  001236                  MOV     FREEZ1,(SP)
  012346  000002                  1$:     RTI
```

```
1 012350                        .ERROR

                                ;ERROR HANDLER

 012350  032777  020000  166422 ERRORS: BIT     @SW13,@SWR                    ; 4
 012356  001055                 BNE     HALTS
 012360  021667  001252         CMP     (SP),LAST
 012364  001404                 BEQ     1$
 012366  011667  001244         MOV     (SP),LAST
 012372  005067  001174         CLR     ERRFLG
 012376  104406                 1$:     SAV05P
 012400  011605                 MOV     (SP),R5
 012402  162705  000002         SUB     #2,R5
 012406  011504                 MOV     (R5),R4
 012410  006304                 ASL     R4
 012412  006304                 ASL     R4
 012414  042704  177001         BIC     #177001,R4
 012420  062704  015216         ADD     #ERRTAB,R4
 012424  012467  000040         MOV     (R4)+,ERRMSG
 012430  011467  000052         MOV     (R4),DATABP
 012434  005767  001132         TST     ERRFLG
 012440  001403                 BEQ     TYPMSG
 012442  005767  000040         TST     DATABP
 012446  001011                 BNE     TYPDAT
 012450  104401                 TYPMSG: TYPE                                  ; 3
 012452  014774                 MCRLF                                         ; 5
 012454  104402                 OCTASC                                        ; 5
 012456  012554                 ERTABO                                        ; 5
 012460  012767  000001  001104 MOV     #1,ERRFLG
 012466  104401                 TYPE
 012470  000000                 ERRMSG: 0
 012472  005767  000010         TYPDAT: TST     DATABP
 012476  001404                 BEQ     RESREG
 012500  104401                 TYPE                                          ; 5
 012502  014774                 MCRLF                                         ; 5
 012504  104402                 OCTASC
 012506  000000                 DATABP: 0
 012510  104407                 RESREG: RES05
 012512  005777  166262         HALTS:  TST     @SWR                          ; 4
 012516  100005                 BPL     EXITER
 012520  010046                 PUSHR0
 012522  016600  000002         MOV     2(SP),R0
 012526  000000                 HALT
 012530  012600                 POPR0
 012532  005267  001040         EXITER: INC     ERRCNT
 012536  032777  002000  166234 BIT     @SW10,@SWR                    ; 4
 012544  001402                 BEQ     1$
 012546  016716  001030         MOV     ESCAPE,(SP)
 012552  000002                 1$:     RTI
 012554  000001                 ERTABO: 1
 012556     006     002         .BYTE   6,2
 012560  013630                 SAVPC
```

```
                                .TRPSRV
                                        ;TRAP DISPATCH SERVICE
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO OBTAIN POINTER                                   ; 3
                                        ;TO SELECTED SUBROUTINE

012562  011646                  TRPSRV: MOV     (SP),-(SP)              ;GET PC OF RETURN
012564  162716  000002                  SUB     #2,(SP)                 ;=PC OF TRAP
012570  017616  000000                  MOV     @(SP),(SP)              ;GET TRP
012574  006316                TRPOK:  ASL     (SP)                    ;MULTIPLY TRAP ARG BY 2
012576  042716  177001                  BIC     #177001,(SP)            ;CLEAR UNWANTED BITS
012602  062716  015136                  ADD     #TRPTAB,(SP)            ;POINTER TO SUBROUTINE ADDRESS
012606  017616  000000                  MOV     @(SP),(SP)              ;SUBROUTINE ADDRESS
012612  000136                        JMP     @(SP)+                  ;GO TO SUBROUTINE
  2 012614                      .SAVREG

                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5

012614  016667  000004  001006  SV05P:  MOV     4(SP),SAVPC

                                        ;SAVE R0-R5

012622  010567  000776           SV05:  MOV     R5,SAVR5
012626  010467  000770                  MOV     R4,SAVR4                                                        ; 3
012632  010367  000762                  MOV     R3,SAVR3
012636  010267  000754                  MOV     R2,SAVR2
012642  010167  000746                  MOV     R1,SAVR1
012646  010067  000740                  MOV     R0,SAVR0
012652  000002                        RTI
  3 012654                      .RESREG
                                        ;RESTORE R0-R5

012654  016700  000732           RS05:  MOV     SAVR0,R0
012660  016701  000730                  MOV     SAVR1,R1
012664  016702  000726                  MOV     SAVR2,R2
012670  016703  000724                  MOV     SAVR3,R3
012674  016704  000722                  MOV     SAVR4,R4
012700  016705  000720                  MOV     SAVR5,R5
012704  000002                        RTI
```

```
      1 012706                          .TYPER

                                        ;TELETYPE OUTPUT ROUTINE

 012706  017605  000000          TYPER:  MOV     @(SP),R5                ; 3
 012712  062716  000002                  ADD     #2,(SP)
 012716  105777  000612          1$:     TSTB    @TPCSR
 012722  100375                          BPL     1$
 012724  105715                          TSTB    (R5)
 012726  001001                          BNE     2$
 012730  000002                          RTI
 012732  112577  000600          2$:     MOVB    (R5)+,@TPDBR
 012736  000767                          BR      1$
      2 012740                          .INSTRG

                                        ;ASCII STRING INPUT ROUTINE

 012740  017667  000000  000006  INSTRG: MOV     @(SP),MSG
 012746  062716  000002                  ADD     #2,(SP)
 012752  104401                  INSTR1: TYPE
 012754  000000                  MSG:    0
 012756  012704  015160                  MOV     #INBUF,R4
 012762  012703  000007                  MOV     #7,R3
 012766  105777  000536          1$:     TSTB    @TKCSR
 012772  100375                          BPL     1$
 012774  117714  000532                  MOVB    @TKDBR,(R4)
 013000  142714  000200                  BICB    #200,(R4)
 013004  122427  000015                  CMPB    (R4)+,#15
 013010  001413                          BEQ     INSTR2
 013012  117777  000514  000516          MOVB    @TKDBR,@TPDBR
 013020  105777  000510          2$:     TSTB    @TPCSR
 013024  100375                          BPL     2$
 013026  005303                          DEC     R3
 013030  001356                          BNE     1$
 013032  104401                  INSTRE: TYPE
 013034  014770                          MQM
 013036  000745                          BR      INSTR1
 013040  000002                  INSTR2: RTI
```

```
1 013042                        .PARAMS

                                        ;CONVERT ASCII STRING TO OCTAL                          ; 3

 013042  011605                 PARAMS: MOV     (SP),R5
 013044  012567  000146                 MOV     (R5)+,LOLIM
 013050  012567  000144                 MOV     (R5)+,HILIM
 013054  012567  000142                 MOV     (R5)+,DEVADR
 013060  112567  000140                 MOVB    (R5)+,LOBITS
 013064  112567  000135                 MOVB    (R5)+,ADRCNT
 013070  010516                         MOV     R5,(SP)
 013072  005005                 PARAM1: CLR     R5
 013074  012704  015160                 MOV     #INBUF,R4
 013100  122714  000015                 CMPB    #15,(R4)
 013104  001420                         BEQ     PARERR
 013106  121427  000060         1$:     CMPB    (R4),#60
 013112  002415                         BLT     PARERR
 013114  121427  000067                 CMPB    (R4),#67
 013120  003012                         BGT     PARERR
 013122  142714  000060                 BICB    #60,(R4)
 013126  152405                         BISB    (R4)+,R5
 013130  122714  000015                 CMPB    #15,(R4)
 013134  001406                         BEQ     LIMITS
 013136  006305                         ASL     R5
 013140  006305                         ASL     R5
 013142  006305                         ASL     R5
 013144  000760                         BR      1$
 013146  104404                 PARERR: INSTER
 013150  000750                         BR      PARAM1

                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

 013152  020567  000042         LIMITS: CMP     R5,HILIM
 013156  101373                         BHI     PARERR
 013160  020567  000032                 CMP     R5,LOLIM
 013164  103770                         BLO     PARERR
 013166  136705  000032                 BITB    LOBITS,R5                                       ; 3
 013172  001365                         BNE     PARERR

                                        ;STORE NUMBER AT SPECIFIED ADDRESS

 013174  016704  000022                 MOV     DEVADR,R4
 013200  010524                 1$:     MOV     R5,(R4)+
 013202  062705  000002                 ADD     #2,R5
 013206  105367  000013                 DECB    ADRCNT
 013212  001372                         BNE     1$
 013214  000002                         RTI
 013216  000000                 LOLIM:  0
 013220  000000                 HILIM:  0
 013222  000000                 DEVADR: 0
 013224  000000                 LOBITS: 0
         013225                 ADRCNT=LOBITS+1
```

```
013226                          .OCTASC

                                ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

013226  017601  000000  OCTASN: MOV     @(SP),R1                        ; 5
013232  062716  000002          ADD     #2,(SP)
013236  012167  000130          MOV     (R1)+,WRDCNT
013242  112167  000126  1$:     MOVB    (R1)+,CHRCNT
013246  112167  000123          MOVB    (R1)+,SPACNT                    ; 3
013252  013167  000120          MOV     @(R1)+,BINWRD
013256  016704  000114  2$:     MOV     BINWRD,R4
013262  116705  000106          MOVB    CHRCNT,R5
013266  012700  015172          MOV     #TEMP,R0
013272  010403          3$:     MOV     R4,R3
013274  042703  177770          BIC     #177770,R3
013300  062703  000260          ADD     #260,R3
013304  110320                  MOVB    R3,(R0)+
013306  006204                  ASR     R4
013310  006204                  ASR     R4
013312  006204                  ASR     R4
013314  005305                  DEC     R5
013316  001365                  BNE     3$
013320  012703  015204          MOV     #MDATA,R3
013324  114023          4$:     MOVB    -(R0),(R3)+
013326  105367  000042          DECB    CHRCNT
013332  001374                  BNE     4$
013334  105767  000035          TSTB    SPACNT
013340  001405                  BEQ     6$
013342  112723  000240  5$:     MOVB    #240,(R3)+
013346  105367  000023          DECB    SPACNT
013352  001373                  BNE     5$
013354  105013          6$:     CLRB    (R3)
013356  104401                  TYPE
013360  015204                  MDATA
013362  005367  000004          DEC     WRDCNT
013366  001325                  BNE     1$
013370  000002                  RTI
013372  000000          WRDCNT: 0
013374  000000          CHRCNT: 0
        013375          SPACNT=CHRCNT+1
013376  000000          BINWRD: 0
```

```
 013400                                 CCLRALL

                                        ;CLEAR ALL BYTE COUNT AND BUS ADDRESS REGISTERS

 013400  012700  000020        CLRALL: MOV     #20,R0          ;SET UP TO CLEAR 16
 013404  005077  000136        1$:     CLR     @DHBA           ;CLEAR BUS ADDRESS
 013410  005077  000134                CLR     @DHBC           ;CLEAR BYTE COUNT
 013414  005277  000120                INC     @DHSCR          ;ANVANCE LINE NUMBER
 013420  005300                        DEC     R0              ;CONTINUE IF NOT DONE
 013422  001370                        BNE     1$
 013424  000207                        RTS     PC              ;RETURN TO CALLING ROUTINE
2 013426                                SSETALL

                                        ;SET BYTE COUNT FOR ALL LINES TO 400
                                        ;SET BUS ADDRESS FOR ALL LINES TO TBUF
                                        ;CLEAR EXPECTED CHARACTER BUFFERS
                                        ;SET LINE ACTIVE BITS FOR ALL LINES

 013426  012700  000020        SETALL: MOV     #20,R0          ;SET UP TO LOAD 16
                                                                ;BYTE COUNT AND BUS ADDRESS
                                                                ;MEMORY LOCATIONS
 013432  005001                        CLR     R1              ;SET UP TO GENERATE EXPECTED
                                                                ;RECEIVED CHARACTER BUFFER
 013434  012702  000200                MOV     #200,R2         ;WILL BE HIGH BYTE
                                                                ;OF EXPECTED RECEIVED CHARACTER
 013440  012703  000001                MOV     #1,R3           ;OFFSET FOR HIGH BYTE
 013444  012777  013646  000074 1$:     MOV     #TBUF,@DHBA     ;LOAD BUS ADDRESS
 013452  012777  177400  000070        MOV     #-400,@DHBC     ;LOAD BYTE COUNT
 013460  012777  031403  000056        MOV     #31403,@DHLPR   ;SET LINE SPEED TO 4800 BAUD
 013466  105061  014246                CLRB    RBUF(R1)
                                                                ;RECEIVED CHARACTER
 013472  110263  014246                MOVB    R2,RBUF(R3)     ;LOAD HIGH BYTE
 013476  005277  000036                INC     @DHSCR          ;ADVANCE LINE NUMBER TO NEXT LINE
 013502  005202                        INC     R2              ;UPDATE POINTERS
 013504  062701  000002                ADD     #2,R1
 013510  062703  000002                ADD     #2,R3
 013514  005300                        DEC     R0              ;CONTINUE IF NOT DONE
 013516  001352                        BNE     1$
 013520  012767  177777  000114        MOV     #-1,LINACT      ;SET ACTIVE FLAGS FOR ALL LINES
 013526  000207                        RTS     PC              ;RETURN TO CALLING ROUTINE
```

```
      1 013530                          .POINT  †/DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DHTLVL/
                                        ;INDIRECT POINTERS                                                              ; 3

        013530  177560                  TKCSR:  177560
        013532  177562                  TKDBR:  177562
        013534  177564                  TPCSR:  177564
        013536  177566                  TPDBR:  177566
                                        .IRP    A       <DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DH
TLVL>
                                        A:      0
                                        .ENDM
        013540  000000                  DHSCR:  0
        013542  000000                  DHNRC:  0
        013544  000000                  DHLPR:  0
        013546  000000                  DHBA:   0
        013550  000000                  DHBC:   0
        013552  000000                  DHBAR:  0
        013554  000000                  DHBCR:  0
        013556  000000                  DHSSR:  0
        013560  000000                  DHSLR:  0
        013562  000000                  DHRVEC: 0
        013564  000000                  DHRLVL: 0
        013566  000000                  DHTVEC: 0
        013570  000000                  DHTLVL: 0
      2 013572                          .VARIA  †/ENDFLG,LINACT,AEACT/
                                        ;PROGRAM VARIABLES

        013572  000000                  ERRFLG: 0               ;ERROR FLAG
        013574  000000                  PASCNT: 0               ;PASS COUNT
        013576  000000                  ERRCNT: 0               ;ERROR COUNT
        013600  000000                  RETRN:  0               ;SCOPE RETURN ADDRESS FOR TEST LOOPING
        013602  000000                  ESCAPE: 0               ;ADDRESS FOR ERROR ESCAPE
        013604  000000                  FREEZ1: 0               ;DATA LOOPING RETURN ADDRESS
        013606  000000                  ICOUNT: 0               ;ITERATION COUNT FOR TEST IN PROGRESS
        013610  000000                  LPCNT:  0               ;NUMBER OF ITERATIONS THIS TEST
        013612  000000                  SAVR0:  0               ;R0 SAVE AREA
        013614  000000                  SAVR1:  0               ;R1 SAVE AREA
        013616  000000                  SAVR2:  0               ;R2 SAVE AREA                                           ; 3
        013620  000000                  SAVR3:  0               ;R3 SAVE ARE
        013622  000000                  SAVR4:  0               ;R4 SAVE AREA
        013624  000000                  SAVR5:  0               ;R5 SAVE AREA
        013626  000000                  SAVSP:  0               ;STACK POINTER SAVE AREA
        013630  000000                  SAVPC:  0               ;CALLING ROUTINE SAVE AREA
        013632  000000                  INIFLG: 0               ;PROGRAM INITIALIZATION FLAG
        013634  000000                  STFLG:  0               ;PROGRAM START FLAG
        013636  000000                  LAST:   0               ;LAST ERROR PC
                                        .IRP    A       <ENDFLG,LINACT,AEACT>
                                        A:      0
                                        .ENDM
        013640  000000                  ENDFLG: 0
        013642  000000                  LINACT: 0
        013644  000000                  AEACT:  0
```

```
                000001                  TDAT=1
      4 013646    000                   TBUF:   .BYTE   0
      5           000377                .REPT   377
      6                                 .BYTE   TDAT
      7                                 .NLIST
      8                                 TDAT=TDAT+1
      9                                 .LIST
     10                                 .ENDR
         013647     001                 .BYTE   TDAT
                000002                  TDAT=TDAT+1
         013650     002                 .BYTE   TDAT
                000003                  TDAT=TDAT+1
         013651     003                 .BYTE   TDAT
                000004                  TDAT=TDAT+1
         013652     004                 .BYTE   TDAT
                000005                  TDAT=TDAT+1
         013653     005                 .BYTE   TDAT
                000006                  TDAT=TDAT+1
         013654     006                 .BYTE   TDAT
                000007                  TDAT=TDAT+1
         013655     007                 .BYTE   TDAT
                000010                  TDAT=TDAT+1
         013656     010                 .BYTE   TDAT
                000011                  TDAT=TDAT+1
         013657     011                 .BYTE   TDAT
                000012                  TDAT=TDAT+1
         013660     012                 .BYTE   TDAT
                000013                  TDAT=TDAT+1
         013661     013                 .BYTE   TDAT
                000014                  TDAT=TDAT+1
         013662     014                 .BYTE   TDAT
                000015                  TDAT=TDAT+1
         013663     015                 .BYTE   TDAT
                000016                  TDAT=TDAT+1
         013664     016                 .BYTE   TDAT
                000017                  TDAT=TDAT+1
         013665     017                 .BYTE   TDAT
                000020                  TDAT=TDAT+1
         013666     020                 .BYTE   TDAT
                000021                  TDAT=TDAT+1
         013667     021                 .BYTE   TDAT
                000022                  TDAT=TDAT+1
         013670     022                 .BYTE   TDAT
                000023                  TDAT=TDAT+1
         013671     023                 .BYTE   TDAT
                000024                  TDAT=TDAT+1
         013672     024                 .BYTE   TDAT
                000025                  TDAT=TDAT+1
         013673     025                 .BYTE   TDAT
                000026                  TDAT=TDAT+1
         013674     026                 .BYTE   TDAT
                000027                  TDAT=TDAT+1
         013675     027                 .BYTE   TDAT
                000030                  TDAT=TDAT+1
         013676     030                 .BYTE   TDAT
                000031                  TDAT=TDAT+1
         013677     031                 .BYTE   TDAT
```

```
                000032                  TDAT=TDAT+1
013700            032                   .BYTE   TDAT
                000033                  TDAT=TDAT+1
013701            033                   .BYTE   TDAT
                000034                  TDAT=TDAT+1
013702            034                   .BYTE   TDAT
                000035                  TDAT=TDAT+1
013703            035                   .BYTE   TDAT
                000036                  TDAT=TDAT+1
013704            036                   .BYTE   TDAT
                000037                  TDAT=TDAT+1
013705            037                   .BYTE   TDAT
                000040                  TDAT=TDAT+1
013706            040                   .BYTE   TDAT
                000041                  TDAT=TDAT+1
013707            041                   .BYTE   TDAT
                000042                  TDAT=TDAT+1
013710            042                   .BYTE   TDAT
                000043                  TDAT=TDAT+1
013711            043                   .BYTE   TDAT
                000044                  TDAT=TDAT+1
013712            044                   .BYTE   TDAT
                000045                  TDAT=TDAT+1
013713            045                   .BYTE   TDAT
                000046                  TDAT=TDAT+1
013714            046                   .BYTE   TDAT
                000047                  TDAT=TDAT+1
013715            047                   .BYTE   TDAT
                000050                  TDAT=TDAT+1
013716            050                   .BYTE   TDAT
                000051                  TDAT=TDAT+1
013717            051                   .BYTE   TDAT
                000052                  TDAT=TDAT+1
013720            052                   .BYTE   TDAT
                000053                  TDAT=TDAT+1
013721            053                   .BYTE   TDAT
                000054                  TDAT=TDAT+1
013722            054                   .BYTE   TDAT
                000055                  TDAT=TDAT+1
013723            055                   .BYTE   TDAT
                000056                  TDAT=TDAT+1
013724            056                   .BYTE   TDAT
                000057                  TDAT=TDAT+1
013725            057                   .BYTE   TDAT
                000060                  TDAT=TDAT+1
013726            060                   .BYTE   TDAT
                000061                  TDAT=TDAT+1
013727            061                   .BYTE   TDAT
                000062                  TDAT=TDAT+1
013730            062                   .BYTE   TDAT
                000063                  TDAT=TDAT+1
013731            063                   .BYTE   TDAT
                000064                  TDAT=TDAT+1
013732            064                   .BYTE   TDAT
                000065                  TDAT=TDAT+1
013733            065                   .BYTE   TDAT
                000066                  TDAT=TDAT+1
```

```
013734    066                 .BYTE   TDAT
          000067              TDAT=TDAT+1
013735    067                 .BYTE   TDAT
          000070              TDAT=TDAT+1
013736    070                 .BYTE   TDAT
          000071              TDAT=TDAT+1
013737    071                 .BYTE   TDAT
          000072              TDAT=TDAT+1
013740    072                 .BYTE   TDAT
          000073              TDAT=TDAT+1
013741    073                 .BYTE   TDAT
          000074              TDAT=TDAT+1
013742    074                 .BYTE   TDAT
          000075              TDAT=TDAT+1
013743    075                 .BYTE   TDAT
          000076              TDAT=TDAT+1
013744    076                 .BYTE   TDAT
          000077              TDAT=TDAT+1
013745    077                 .BYTE   TDAT
          000100              TDAT=TDAT+1
013746    100                 .BYTE   TDAT
          000101              TDAT=TDAT+1
013747    101                 .BYTE   TDAT
          000102              TDAT=TDAT+1
013750    102                 .BYTE   TDAT
          000103              TDAT=TDAT+1
013751    103                 .BYTE   TDAT
          000104              TDAT=TDAT+1
013752    104                 .BYTE   TDAT
          000105              TDAT=TDAT+1
013753    105                 .BYTE   TDAT
          000106              TDAT=TDAT+1
013754    106                 .BYTE   TDAT
          000107              TDAT=TDAT+1
013755    107                 .BYTE   TDAT
          000110              TDAT=TDAT+1
013756    110                 .BYTE   TDAT
          000111              TDAT=TDAT+1
013757    111                 .BYTE   TDAT
          000112              TDAT=TDAT+1
013760    112                 .BYTE   TDAT
          000113              TDAT=TDAT+1
013761    113                 .BYTE   TDAT
          000114              TDAT=TDAT+1
013762    114                 .BYTE   TDAT
          000115              TDAT=TDAT+1
013763    115                 .BYTE   TDAT
          000116              TDAT=TDAT+1
013764    116                 .BYTE   TDAT
          000117              TDAT=TDAT+1
013765    117                 .BYTE   TDAT
          000120              TDAT=TDAT+1
013766    120                 .BYTE   TDAT
          000121              TDAT=TDAT+1
013767    121                 .BYTE   TDAT
          000122              TDAT=TDAT+1
013770    122                 .BYTE   TDAT
```

```
                000123                  TDAT=TDAT+1
013771             123                  .BYTE   TDAT
                000124                  TDAT=TDAT+1
013772             124                  .BYTE   TDAT
                000125                  TDAT=TDAT+1
013773             125                  .BYTE   TDAT
                000126                  TDAT=TDAT+1
013774             126                  .BYTE   TDAT
                000127                  TDAT=TDAT+1
013775             127                  .BYTE   TDAT
                000130                  TDAT=TDAT+1
013776             130                  .BYTE   TDAT
                000131                  TDAT=TDAT+1
013777             131                  .BYTE   TDAT
                000132                  TDAT=TDAT+1
014000             132                  .BYTE   TDAT
                000133                  TDAT=TDAT+1
014001             133                  .BYTE   TDAT
                000134                  TDAT=TDAT+1
014002             134                  .BYTE   TDAT
                000135                  TDAT=TDAT+1
014003             135                  .BYTE   TDAT
                000136                  TDAT=TDAT+1
014004             136                  .BYTE   TDAT
                000137                  TDAT=TDAT+1
014005             137                  .BYTE   TDAT
                000140                  TDAT=TDAT+1
014006             140                  .BYTE   TDAT
                000141                  TDAT=TDAT+1
014007             141                  .BYTE   TDAT
                000142                  TDAT=TDAT+1
014010             142                  .BYTE   TDAT
                000143                  TDAT=TDAT+1
014011             143                  .BYTE   TDAT
                000144                  TDAT=TDAT+1
014012             144                  .BYTE   TDAT
                000145                  TDAT=TDAT+1
014013             145                  .BYTE   TDAT
                000146                  TDAT=TDAT+1
014014             146                  .BYTE   TDAT
                000147                  TDAT=TDAT+1
014015             147                  .BYTE   TDAT
                000150                  TDAT=TDAT+1
014016             150                  .BYTE   TDAT
                000151                  TDAT=TDAT+1
014017             151                  .BYTE   TDAT
                000152                  TDAT=TDAT+1
014020             152                  .BYTE   TDAT
                000153                  TDAT=TDAT+1
014021             153                  .BYTE   TDAT
                000154                  TDAT=TDAT+1
014022             154                  .BYTE   TDAT
                000155                  TDAT=TDAT+1
014023             155                  .BYTE   TDAT
                000156                  TDAT=TDAT+1
014024             156                  .BYTE   TDAT
                000157                  TDAT=TDAT+1
```

```
014025    157                 .BYTE   TDAT
          000160              TDAT=TDAT+1
014026    160                 .BYTE   TDAT
          000161              TDAT=TDAT+1
014027    161                 .BYTE   TDAT
          000162              TDAT=TDAT+1
014030    162                 .BYTE   TDAT
          000163              TDAT=TDAT+1
014031    163                 .BYTE   TDAT
          000164              TDAT=TDAT+1
014032    164                 .BYTE   TDAT
          000165              TDAT=TDAT+1
014033    165                 .BYTE   TDAT
          000166              TDAT=TDAT+1
014034    166                 .BYTE   TDAT
          000167              TDAT=TDAT+1
014035    167                 .BYTE   TDAT
          000170              TDAT=TDAT+1
014036    170                 .BYTE   TDAT
          000171              TDAT=TDAT+1
014037    171                 .BYTE   TDAT
          000172              TDAT=TDAT+1
014040    172                 .BYTE   TDAT
          000173              TDAT=TDAT+1
014041    173                 .BYTE   TDAT
          000174              TDAT=TDAT+1
014042    174                 .BYTE   TDAT
          000175              TDAT=TDAT+1
014043    175                 .BYTE   TDAT
          000176              TDAT=TDAT+1
014044    176                 .BYTE   TDAT
          000177              TDAT=TDAT+1
014045    177                 .BYTE   TDAT
          000200              TDAT=TDAT+1
014046    200                 .BYTE   TDAT
          000201              TDAT=TDAT+1
014047    201                 .BYTE   TDAT
          000202              TDAT=TDAT+1
014050    202                 .BYTE   TDAT
          000203              TDAT=TDAT+1
014051    203                 .BYTE   TDAT
          000204              TDAT=TDAT+1
014052    204                 .BYTE   TDAT
          000205              TDAT=TDAT+1
014053    205                 .BYTE   TDAT
          000206              TDAT=TDAT+1
014054    206                 .BYTE   TDAT
          000207              TDAT=TDAT+1
014055    207                 .BYTE   TDAT
          000210              TDAT=TDAT+1
014056    210                 .BYTE   TDAT
          000211              TDAT=TDAT+1
014057    211                 .BYTE   TDAT
          000212              TDAT=TDAT+1
014060    212                 .BYTE   TDAT
          000213              TDAT=TDAT+1
014061    213                 .BYTE   TDAT
```

```
                000214                  TDAT=TDAT+1
014062             214                  .BYTE   TDAT
                000215                  TDAT=TDAT+1
014063             215                  .BYTE   TDAT
                000216                  TDAT=TDAT+1
014064             216                  .BYTE   TDAT
                000217                  TDAT=TDAT+1
014065             217                  .BYTE   TDAT
                000220                  TDAT=TDAT+1
014066             220                  .BYTE   TDAT
                000221                  TDAT=TDAT+1
014067             221                  .BYTE   TDAT
                000222                  TDAT=TDAT+1
014070             222                  .BYTE   TDAT
                000223                  TDAT=TDAT+1
014071             223                  .BYTE   TDAT
                000224                  TDAT=TDAT+1
014072             224                  .BYTE   TDAT
                000225                  TDAT=TDAT+1
014073             225                  .BYTE   TDAT
                000226                  TDAT=TDAT+1
014074             226                  .BYTE   TDAT
                000227                  TDAT=TDAT+1
014075             227                  .BYTE   TDAT
                000230                  TDAT=TDAT+1
014076             230                  .BYTE   TDAT
                000231                  TDAT=TDAT+1
014077             231                  .BYTE   TDAT
                000232                  TDAT=TDAT+1
014100             232                  .BYTE   TDAT
                000233                  TDAT=TDAT+1
014101             233                  .BYTE   TDAT
                000234                  TDAT=TDAT+1
014102             234                  .BYTE   TDAT
                000235                  TDAT=TDAT+1
014103             235                  .BYTE   TDAT
                000236                  TDAT=TDAT+1
014104             236                  .BYTE   TDAT
                000237                  TDAT=TDAT+1
014105             237                  .BYTE   TDAT
                000240                  TDAT=TDAT+1
014106             240                  .BYTE   TDAT
                000241                  TDAT=TDAT+1
014107             241                  .BYTE   TDAT
                000242                  TDAT=TDAT+1
014110             242                  .BYTE   TDAT
                000243                  TDAT=TDAT+1
014111             243                  .BYTE   TDAT
                000244                  TDAT=TDAT+1
014112             244                  .BYTE   TDAT
                000245                  TDAT=TDAT+1
014113             245                  .BYTE   TDAT
                000246                  TDAT=TDAT+1
014114             246                  .BYTE   TDAT
                000247                  TDAT=TDAT+1
014115             247                  .BYTE   TDAT
                000250                  TDAT=TDAT+1
```

```
014116     250                  .BYTE   TDAT
           000251               TDAT=TDAT+1
014117     251                  .BYTE   TDAT
           000252               TDAT=TDAT+1
014120     252                  .BYTE   TDAT
           000253               TDAT=TDAT+1
014121     253                  .BYTE   TDAT
           000254               TDAT=TDAT+1
014122     254                  .BYTE   TDAT
           000255               TDAT=TDAT+1
014123     255                  .BYTE   TDAT
           000256               TDAT=TDAT+1
014124     256                  .BYTE   TDAT
           000257               TDAT=TDAT+1
014125     257                  .BYTE   TDAT
           000260               TDAT=TDAT+1
014126     260                  .BYTE   TDAT
           000261               TDAT=TDAT+1
014127     261                  .BYTE   TDAT
           000262               TDAT=TDAT+1
014130     262                  .BYTE   TDAT
           000263               TDAT=TDAT+1
014131     263                  .BYTE   TDAT
           000264               TDAT=TDAT+1
014132     264                  .BYTE   TDAT
           000265               TDAT=TDAT+1
014133     265                  .BYTE   TDAT
           000266               TDAT=TDAT+1
014134     266                  .BYTE   TDAT
           000267               TDAT=TDAT+1
014135     267                  .BYTE   TDAT
           000270               TDAT=TDAT+1
014136     270                  .BYTE   TDAT
           000271               TDAT=TDAT+1
014137     271                  .BYTE   TDAT
           000272               TDAT=TDAT+1
014140     272                  .BYTE   TDAT
           000273               TDAT=TDAT+1
014141     273                  .BYTE   TDAT
           000274               TDAT=TDAT+1
014142     274                  .BYTE   TDAT
           000275               TDAT=TDAT+1
014143     275                  .BYTE   TDAT
           000276               TDAT=TDAT+1
014144     276                  .BYTE   TDAT
           000277               TDAT=TDAT+1
014145     277                  .BYTE   TDAT
           000300               TDAT=TDAT+1
014146     300                  .BYTE   TDAT
           000301               TDAT=TDAT+1
014147     301                  .BYTE   TDAT
           000302               TDAT=TDAT+1
014150     302                  .BYTE   TDAT
           000303               TDAT=TDAT+1
014151     303                  .BYTE   TDAT
           000304               TDAT=TDAT+1
014152     304                  .BYTE   TDAT
```

```
                000305                  TDAT=TDAT+1
014153             305                  .BYTE   TDAT
                000306                  TDAT=TDAT+1
014154             306                  .BYTE   TDAT
                000307                  TDAT=TDAT+1
014155             307                  .BYTE   TDAT
                000310                  TDAT=TDAT+1
014156             310                  .BYTE   TDAT
                000311                  TDAT=TDAT+1
014157             311                  .BYTE   TDAT
                000312                  TDAT=TDAT+1
014160             312                  .BYTE   TDAT
                000313                  TDAT=TDAT+1
014161             313                  .BYTE   TDAT
                000314                  TDAT=TDAT+1
014162             314                  .BYTE   TDAT
                000315                  TDAT=TDAT+1
014163             315                  .BYTE   TDAT
                000316                  TDAT=TDAT+1
014164             316                  .BYTE   TDAT
                000317                  TDAT=TDAT+1
014165             317                  .BYTE   TDAT
                000320                  TDAT=TDAT+1
014166             320                  .BYTE   TDAT
                000321                  TDAT=TDAT+1
014167             321                  .BYTE   TDAT
                000322                  TDAT=TDAT+1
014170             322                  .BYTE   TDAT
                000323                  TDAT=TDAT+1
014171             323                  .BYTE   TDAT
                000324                  TDAT=TDAT+1
014172             324                  .BYTE   TDAT
                000325                  TDAT=TDAT+1
014173             325                  .BYTE   TDAT
                000326                  TDAT=TDAT+1
014174             326                  .BYTE   TDAT
                000327                  TDAT=TDAT+1
014175             327                  .BYTE   TDAT
                000330                  TDAT=TDAT+1
014176             330                  .BYTE   TDAT
                000331                  TDAT=TDAT+1
014177             331                  .BYTE   TDAT
                000332                  TDAT=TDAT+1
014200             332                  .BYTE   TDAT
                000333                  TDAT=TDAT+1
014201             333                  .BYTE   TDAT
                000334                  TDAT=TDAT+1
014202             334                  .BYTE   TDAT
                000335                  TDAT=TDAT+1
014203             335                  .BYTE   TDAT
                000336                  TDAT=TDAT+1
014204             336                  .BYTE   TDAT
                000337                  TDAT=TDAT+1
014205             337                  .BYTE   TDAT
                000340                  TDAT=TDAT+1
014206             340                  .BYTE   TDAT
                000341                  TDAT=TDAT+1
```

```
014207    341                 .BYTE   TDAT
          000342              TDAT=TDAT+1
014210    342                 .BYTE   TDAT
          000343              TDAT=TDAT+1
014211    343                 .BYTE   TDAT
          000344              TDAT=TDAT+1
014212    344                 .BYTE   TDAT
          000345              TDAT=TDAT+1
014213    345                 .BYTE   TDAT
          000346              TDAT=TDAT+1
014214    346                 .BYTE   TDAT
          000347              TDAT=TDAT+1
014215    347                 .BYTE   TDAT
          000350              TDAT=TDAT+1
014216    350                 .BYTE   TDAT
          000351              TDAT=TDAT+1
014217    351                 .BYTE   TDAT
          000352              TDAT=TDAT+1
014220    352                 .BYTE   TDAT
          000353              TDAT=TDAT+1
014221    353                 .BYTE   TDAT
          000354              TDAT=TDAT+1
014222    354                 .BYTE   TDAT
          000355              TDAT=TDAT+1
014223    355                 .BYTE   TDAT
          000356              TDAT=TDAT+1
014224    356                 .BYTE   TDAT
          000357              TDAT=TDAT+1
014225    357                 .BYTE   TDAT
          000360              TDAT=TDAT+1
014226    360                 .BYTE   TDAT
          000361              TDAT=TDAT+1
014227    361                 .BYTE   TDAT
          000362              TDAT=TDAT+1
014230    362                 .BYTE   TDAT
          000363              TDAT=TDAT+1
014231    363                 .BYTE   TDAT
          000364              TDAT=TDAT+1
014232    364                 .BYTE   TDAT
          000365              TDAT=TDAT+1
014233    365                 .BYTE   TDAT
          000366              TDAT=TDAT+1
014234    366                 .BYTE   TDAT
          000367              TDAT=TDAT+1
014235    367                 .BYTE   TDAT
          000370              TDAT=TDAT+1
014236    370                 .BYTE   TDAT
          000371              TDAT=TDAT+1
014237    371                 .BYTE   TDAT
          000372              TDAT=TDAT+1
014240    372                 .BYTE   TDAT
          000373              TDAT=TDAT+1
014241    373                 .BYTE   TDAT
          000374              TDAT=TDAT+1
014242    374                 .BYTE   TDAT
          000375              TDAT=TDAT+1
014243    375                 .BYTE   TDAT
```

```
            000376                  TDAT=TDAT+1
    014244     376                  .BYTE   TDAT
            000377                  TDAT=TDAT+1
    014245     377                  .BYTE   TDAT
            000400                  TDAT=TDAT+1
11                                  .EVEN
12  014246  000000                  RBUF:   0
13          014310                  .=.+40
```

```
 1                              .MACRO  WORDS   WDNAME,K,DATA
 2                              'WDNAME'K':     DATA
 3                              .ENDM   WORDS
 5          000020              K=KX
 6          000000              DATA=DATAX
 7          000000              KX=0
 9          000020              .REPT   20
10                              .NLIST
11                              DATAX=KX*400+100377
12                              .LIST
13                              WORDS   \/TWRD/,\KX,\DATAX
14                              .NLIST
15                              KX=KX+1
16                              .LIST
17                              .ENDR
           100377               DATAX=KX*400+100377
    014310                      WORDS   \/TWRD/,\KX,\DATAX
    014310  100377              TWRD0:  100377
           000001               KX=KX+1
           100777               DATAX=KX*400+100377
    014312                      WORDS   \/TWRD/,\KX,\DATAX
    014312  100777              TWRD1:  100777
           000002               KX=KX+1
           101377               DATAX=KX*400+100377
    014314                      WORDS   \/TWRD/,\KX,\DATAX
    014314  101377              TWRD2:  101377
           000003               KX=KX+1
           101777               DATAX=KX*400+100377
    014316                      WORDS   \/TWRD/,\KX,\DATAX
    014316  101777              TWRD3:  101777
           000004               KX=KX+1
           102377               DATAX=KX*400+100377
    014320                      WORDS   \/TWRD/,\KX,\DATAX
    014320  102377              TWRD4:  102377
           000005               KX=KX+1
           102777               DATAX=KX*400+100377
    014322                      WORDS   \/TWRD/,\KX,\DATAX
    014322  102777              TWRD5:  102777
           000006               KX=KX+1
           103377               DATAX=KX*400+100377
    014324                      WORDS   \/TWRD/,\KX,\DATAX
    014324  103377              TWRD6:  103377
           000007               KX=KX+1
           103777               DATAX=KX*400+100377
    014326                      WORDS   \/TWRD/,\KX,\DATAX
    014326  103777              TWRD7:  103777
           000010               KX=KX+1
           104377               DATAX=KX*400+100377
    014330                      WORDS   \/TWRD/,\KX,\DATAX
    014330  104377              TWRD10: 104377
           000011               KX=KX+1
           104777               DATAX=KX*400+100377
    014332                      WORDS   \/TWRD/,\KX,\DATAX
    014332  104777              TWRD11: 104777
           000012               KX=KX+1
           105377               DATAX=KX*400+100377
    014334                      WORDS   \/TWRD/,\KX,\DATAX
```

```
014334  105377          TWRD12: 105377
        000013          KX=KX+1
        105777          DATAX=KX*400+100377
014336                  WORDS   +/TWRD/,\KX,\DATAX
014336  105777          TWRD13: 105777
        000014          KX=KX+1
        106377          DATAX=KX*400+100377
014340                  WORDS   +/TWRD/,\KX,\DATAX
014340  106377          TWRD14: 106377
        000015          KX=KX+1
        106777          DATAX=KX*400+100377
014342                  WORDS   +/TWRD/,\KX,\DATAX
014342  106777          TWRD15: 106777
        000016          KX=KX+1
        107377          DATAX=KX*400+100377
014344                  WORDS   +/TWRD/,\KX,\DATAX
014344  107377          TWRD16: 107377
        000017          KX=KX+1
        107777          DATAX=KX*400+100377
014346                  WORDS   +/TWRD/,\KX,\DATAX
014346  107777          TWRD17: 107777
        000020          KX=KX+1
   19   107777          DATA=DATAX
   20   000020          K=KX
   21   000000          DATAX=0
   22   000000          KX=0
   24   000020          .REPT   20
   25                   WORDS   +/RCNT/,\KX,\DATAX
   26                   .NLIST
   27                   KX=KX+1
   28                   .LIST
   29                   .ENDR
014350                  WORDS   +/RCNT/,\KX,\DATAX
014350  000000          RCNT0:  0
        000001          KX=KX+1
014352                  WORDS   +/RCNT/,\KX,\DATAX
014352  000000          RCNT1:  0
        000002          KX=KX+1
014354                  WORDS   +/RCNT/,\KX,\DATAX
014354  000000          RCNT2:  0
        000003          KX=KX+1
014356                  WORDS   +/RCNT/,\KX,\DATAX
014356  000000          RCNT3:  0
        000004          KX=KX+1
014360                  WORDS   +/RCNT/,\KX,\DATAX
014360  000000          RCNT4:  0
        000005          KX=KX+1
014362                  WORDS   +/RCNT/,\KX,\DATAX
014362  000000          RCNT5:  0
        000006          KX=KX+1
014364                  WORDS   +/RCNT/,\KX,\DATAX
014364  000000          RCNT6:  0
        000007          KX=KX+1
014366                  WORDS   +/RCNT/,\KX,\DATAX
014366  000000          RCNT7:  0
        000010          KX=KX+1
014370                  WORDS   +/RCNT/,\KX,\DATAX
```

```
014370  000000          RCNT10: 0
        000011          KX=KX+1
014372                  WORDS   +/RCNT/,\KX,\DATAX
014372  000000          RCNT11: 0
        000012          KX=KX+1
014374                  WORDS   +/RCNT/,\KX,\DATAX
014374  000000          RCNT12: 0
        000013          KX=KX+1
014376                  WORDS   +/RCNT/,\KX,\DATAX
014376  000000          RCNT13: 0
        000014          KX=KX+1
014400                  WORDS   +/RCNT/,\KX,\DATAX
014400  000000          RCNT14: 0
        000015          KX=KX+1
014402                  WORDS   +/RCNT/,\KX,\DATAX
014402  000000          RCNT15: 0
        000016          KX=KX+1
014404                  WORDS   +/RCNT/,\KX,\DATAX
014404  000000          RCNT16: 0
        000017          KX=KX+1
014406                  WORDS   +/RCNT/,\KX,\DATAX
014406  000000          RCNT17: 0
        000020          KX=KX+1
31      000000          DATA=DATAX
32      000020          K=KX
33      000000          DATAX=0
34      000000          KX=0
36      000020          .REPT   20
37                      WORDS   +/RDCT/,\KX,\DATAX
38                      .NLIST
39                      KX=KX+1
40                      .LIST
41                      .ENDR
014410                  WORDS   +/RDCT/,\KX,\DATAX
014410  000000          RDCT0:  0
        000001          KX=KX+1
014412                  WORDS   +/RDCT/,\KX,\DATAX
014412  000000          RDCT1:  0
        000002          KX=KX+1
014414                  WORDS   +/RDCT/,\KX,\DATAX
014414  000000          RDCT2:  0
        000003          KX=KX+1
014416                  WORDS   +/RDCT/,\KX,\DATAX
014416  000000          RDCT3:  0
        000004          KX=KX+1
014420                  WORDS   +/RDCT/,\KX,\DATAX
014420  000000          RDCT4:  0
        000005          KX=KX+1
014422                  WORDS   +/RDCT/,\KX,\DATAX
014422  000000          RDCT5:  0
        000006          KX=KX+1
014424                  WORDS   +/RDCT/,\KX,\DATAX
014424  000000          RDCT6:  0
        000007          KX=KX+1
014426                  WORDS   +/RDCT/,\KX,\DATAX
014426  000000          RDCT7:  0
        000010          KX=KX+1
```

```
                014430                          WORDS   +/RDCT/,\KX,\DATAX
                014430  000000                  RDCT10: 0
                        000011                  KX=KX+1
                014432                          WORDS   +/RDCT/,\KX,\DATAX
                014432  000000                  RDCT11: 0
                        000012                  KX=KX+1
                014434                          WORDS   +/RDCT/,\KX,\DATAX
                014434  000000                  RDCT12: 0
                        000013                  KX=KX+1
                014436                          WORDS   +/RDCT/,\KX,\DATAX
                014436  000000                  RDCT13: 0
                        000014                  KX=KX+1
                014440                          WORDS   +/RDCT/,\KX,\DATAX
                014440  000000                  RDCT14: 0
                        000015                  KX=KX+1
                014442                          WORDS   +/RDCT/,\KX,\DATAX
                014442  000000                  RDCT15: 0
                        000016                  KX=KX+1
                014444                          WORDS   +/RDCT/,\KX,\DATAX
                014444  000000                  RDCT16: 0
                        000017                  KX=KX+1
                014446                          WORDS   +/RDCT/,\KX,\DATAX
                014446  000000                  RDCT17: 0
                        000020                  KX=KX+1
             42 014450  000001                  LINBIT: 1
             43 014452  000002                          2
             44 014454  000004                          4
             45 014456  000010                          10
             46 014460  000020                          20
             47 014462  000040                          40
             48 014464  000100                          100
             49 014466  000200                          200
             50 014470  000400                          400
             51 014472  001000                          1000
             52 014474  002000                          2000
             53 014476  004000                          4000
             54 014500  010000                          10000
             55 014502  020000                          20000
             56 014504  040000                          40000
             57 014506  100000                          100000
```

```
1 014510                        .PFAIL
                                ;ENTER HERE ON POWER FAILURE

 014510  010046                 PFAIL:  MOV     R0,-(SP)        ;SAVE R0-R5 ON PROCESSOR STACK
 014512  010146                         MOV     R1,-(SP)
 014514  010246                         MOV     R2,-(SP)
 014516  010346                         MOV     R3,-(SP)
 014520  010446                         MOV     R4,-(SP)
 014522  010546                         MOV     R5,-(SP)
 014524  016746  163274                 MOV     24,-(SP)
 014530  010667  177072                 MOV     SP,SAVSP        ;SAVE STACK POINTER
 014534  012767  014546  163262         MOV     #RESTART,24     ;SET UP FOR POWER UP TRAP       ; 3
 014542  000000                         HALT                    ;HALT ON POWER DOWN NORMAL
 014544  000777                         BR      .

                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

 014546  016706  177054         RESTAR: MOV     SAVSP,SP        ;RESTORE STACK POINTER
 014552  012605                         MOV     (SP)+,R5        ;RESTORE R0-R5
 014554  012604                         MOV     (SP)+,R4
 014556  012603                         MOV     (SP)+,R3
 014560  012602                         MOV     (SP)+,R2
 014562  012601                         MOV     (SP)+,R1
 014564  012600                         MOV     (SP)+,R0
 014566  012767  014510  163230         MOV     #PFAIL,24       ;SET UP FOR POWER FAILURE
 014574  012767  000340  163174         MOV     #340,PS
 014602  012706  015644                 MOV     #STACK,SP
 014606  005067  000360                 CLR     TEMP
 014612  005267  000354                 INC     TEMP
 014616  001375                         BNE     .-4
 014620  104401                         TYPE                                            ; 5
 014622  014774                         MCRLF                                           ; 5
 014624  104402                         OCTASC
 014626  014650                         PFTAB
 014630  104401                         TYPE
 014632  014777                         MPFAIL
 014634  005067  176732                 CLR     ERRFLG
 014640  005067  176772                 CLR     LAST
 014644  000177  176730                 JMP     @RETRN
 014650  000001          PFTAB:  1
 014652  000006  000002                 6,2
 014656  013600                         RETRN
```

```
014660                                  .MSG    †/DH11 AUTO ECHO TEST/,†/CZDHH-CO/
014660  015  012  012  MTITLE: .ASCIZ  <15><12><12>/DH11 AUTO ECHO TEST /<15><12>
014663  104  110  061
014666  061  040  101
014671  125  124  117
014674  040  105  103
014677  110  117  040
014702  124  105  123
014705  124  040  015
014710  012  000
014712  015  012  126  MVECTO: .ASCIZ  <15><12>/VECTOR ADDRESS-/
014715  105  103  124
014720  117  122  040
014723  101  104  104
014726  122  105  123
014731  123  055  000
014734  015  012  103  MREGAD: .ASCIZ  <15><12>/CONTROL REGISTER ADDRESS-/
014737  117  116  124
014742  122  117  114
014745  040  122  105
014750  107  111  123
014753  124  105  122
014756  040  101  104
014761  104  122  105
014764  123  123  055
014767  000
014770  040  040  077  MQM:    .ASCIZ  /  ?/
014773  000
014774  015  012  000  MCRLF:  .ASCIZ   <15><12>

014777  040  040  120  MPFAIL: .ASCIZ  /  POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
015002  117  127  105
015005  122  040  106
015010  101  111  114
015013  125  122  105
015016  054  040  120
015021  122  117  107
015024  122  101  115
015027  040  122  105
015032  123  124  101
015035  122  124  040
015040  101  124  040
015043  124  105  123
015046  124  040  111
015051  116  040  120
015054  122  117  107
015057  122  105  123
015062  123  000
015064  015  012  103  MEPASS: .ASCIZ  <15><12>/CZDHH-CO/
015067  132  104  110
015072  110  055  103
015075  060  000
015077  015  012  120  PASTXT: .ASCIZ  <15><12>/PASS COUNT = /                    ; 5
015102  101  123  123
015105  040  103  117
015110  125  116  124
015113  040  075  040
```

```
                015116     000
                015117     015     012     122  MR:      .ASCIZ  <15><12>/R/
                015122     000
                015123     015     012     124  MTSTPC:  .ASCIZ  <15><12>/TEST PC-/
                015126     105     123     124
                015131     040     120     103
                015134     055     000

                                                .EVEN
                                                .EVEN
          3     015136                          .TRPTAB

                                                        ;TABLE OF POINTERS FOR TRAP DECODING

                015136  012232                  TRPTAB: SCOPER
                015140  012706                          TYPER
                015142  013226                          OCTASN
                015144  012740                          INSTRG
                015146  013032                          INSTRE
                015150  013042                          PARAMS
                015152  012614                          SV05P
                015154  012654                          RS05
                015156  012332                          SCOP1R
          4     015160                          .BUFFER

                                                        ;BUFFERS FOR INPUT-OUTPUT

                015160  000000                  INBUF:  0
                        015172                  .=.+10
                015172  000000                  TEMP:   0
                        015204                  .=.+10

                015204  000000                  MDATA:  0
                        015216                  .=.+10
          5     015216                          .ERRTAB

                                                        ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA

                015216                          ERRTAB:
          6     015216  015232                          EM1
          7     015220  015414                          DT1
          8     015222  015267                          EM2
          9     015224  015426                          DT2
         10     015226  015343                          EM3
         11     015230  015426                          DT2
         12     015232     101     125     124  EM1:     .ASCIZ  /AUTO ECHO ERROR/<15><12>/EXP     REC/
                015235     117     040     105
                015240     103     110     117
                015243     040     105     122
                015246     122     117     122
                015251     015     012     105
                015254     130     120     040
                015257     040     040     040
                015262     040     122     105
                015265     103     000
         13     015267     116     117     116  EM2:     .ASCIZ  /NON ECHOED DATA ERROR/<15><12>/EXP     REC     LINE/
                015272     040     105     103
                015275     110     117     105
```

```
         015300    104    040    104
         015303    101    124    101
         015306    040    105    122
         015311    122    117    122
         015314    015    012    105
         015317    130    120    040
         015322    040    040    040
         015325    040    122    105
         015330    103    040    040
         015333    040    040    040
         015336    114    111    116
         015341    105    000
   14    015343    105    103    110  EM3:    .ASCIZ  /ECHOED DATA ERROR/<15><12>/EXP     REC     LINE/
         015346    117    105    104
         015351    040    104    101
         015354    124    101    040
         015357    105    122    122
         015362    117    122    015
         015365    012    105    130
         015370    120    040    040
         015373    040    040    040
         015376    122    105    103
         015401    040    040    040
         015404    040    040    114
         015407    111    116    105
         015412    000
   15                              .EVEN
   16    015414  000002            DT1:    2
   17    015416     006    002     .BYTE   6,2
   18    015420  013624                    SAVR5
   19    015422     006    000     .BYTE   6,0
   20    015424  013622                    SAVR4
   21    015426  000003            DT2:    3
   22    015430     006    002     .BYTE   6,2
   23    015432  013624                    SAVR5
   24    015434     006    002     .BYTE   6,2
   25    015436  013622                    SAVR4
   26    015440     002    002     .BYTE   2,2
   27    015442  013620                    SAVR3
   28    015444                    .ENDCOD
         015444  000000            ENDCOD: 0
   29            000001            .END
```

```
CZDHH-CO        MACRO V04.00  27-JUN-85 13:04:20 PAGE 36 3
SYMBOL TABLE

ADRCNT- 013225          ERRTAB  015216          POPRO - 012600          SAVR4   013622          TWRD2   014314
AEACT   013644          ERTABO  012554          POP1SP- 005726          SAVR5   013624          TWRD3   014316
BEGIN   001306          ESCAPE  013602          POP2SP- 022626          SAVSP   013626          TWRD4   014320
BINWRD  013376          EXITER  012532          PS      - 177776        SAVOSP- 104406          TWRD5   014322
BITX  - 000000          FREEZ1  013604          PUSHRO- 010046          SCOPE - 104400          TWRD6   014324
BIT00 - 000001          HALTS   012512          PUSH1S- 005746          SCOPER  012232          TWRD7   014326
BIT01 - 000002          HILIM   013220          PUSH2S- 024646          SCOPE1- 104410          TYPDAT  012472
BIT02 - 000004          ICOUNT  013606          RBUF    014246          SCOP1R  012332          TYPE  - 104401
BIT03 - 000010          INBUF   015160          RCNT0   014350          SETALL  013426          TYPER   012706
BIT04 - 000020          INIFLG  013632          RCNT1   014352          SPACNT- 013375          TYPMSG  012450
BIT05 - 000040          INSTER- 104404          RCNT10  014370          STACK - 015644          T1      001400
BIT06 - 000100          INSTR - 104403          RCNT11  014372          START   001004          T10     003002
BIT07 - 000200          INSTRE  013032          RCNT12  014374          STFLG   013634          T11     003160
BIT08 - 000400          INSTRG  012740          RCNT13  014376          SV05    012622          T12     003336
BIT09 - 001000          INSTR1  012752          RCNT14  014400          SV05P   012614          T13     003514
BIT10 - 002000          INSTR2  013040          RCNT15  014402          SWR     001000          T14     003672
BIT11 - 004000          K     - 000020          RCNT16  014404          SW00  - 000001          T15     004050
BIT12 - 010000          KX    - 000020          RCNT17  014406          SW01  - 000002          T16     004226
BIT13 - 020000          LAST    013636          RCNT2   014354          SW02  - 000004          T17     004404
BIT14 - 040000          LIGHTS  001002          RCNT3   014356          SW03  - 000010          T2      001556
BIT15 - 100000          LIMITS  013152          RCNT4   014360          SW04  - 000020          T20     004562
CHRCNT  013374          LINACT  013642          RCNT5   014362          SW05  - 000040          T21     004740
CLRALL  013400          LINBIT  014450          RCNT6   014364          SW06  - 000100          T22     005176
DATA  - 000000          LINE  - 000020          RCNT7   014366          SW08  - 000400          T23     005434
DATABP  012506          LOBITS  013224          RDCT0   014410          SW09  - 001000          T24     005672
DATAX - 000000          LOGICA  012210          RDCT1   014412          SW10  - 002000          T25     006130
DEVADR  013222          LOLIM   013216          RDCT10  014430          SW11  - 004000          T26     006366
DHBA    013546          LPCNT   013610          RDCT11  014432          SW12  - 010000          T27     006624
DHBAR   013552          MCRLF   014774          RDCT12  014434          SW13  - 020000          T3      001734
DHBC    013550          MDATA   015204          RDCT13  014436          SW14  - 040000          T30     007062
DHBCR   013554          MEPASS  015064          RDCT14  014440          SW15  - 100000          T31     007320
DHLPR   013544          MPFAIL  014777          RDCT15  014442          TBUF    013646          T32     007556
DHNRC   013542          MQM     014770          RDCT16  014444          TDAT  - 000400          T33     010014
DHRLVL  013564          MR      015117          RDCT17  014446          TEMP    015172          T34     010252
DHRVEC  013562          MREGAD  014734          RDCT2   014414          TKCSR   013530          T35     010510
DHSCR   013540          MSG     012754          RDCT3   014416          TKDBR   013532          T36     010746
DHSLR   013560          MTITLE  014660          RDCT4   014420          TPCSR   013534          T37     011204
DHSSR   013556          MTSTPC  015123          RDCT5   014422          TPDBR   013536          T4      002112
DHTLVL  013570          MVECTO  014712          RDCT6   014424          TRPOK   012574          T40     011442
DHTVEC  013566          N     - 000001          RDCT7   014426          TRPSRV  012562          T41     011700
DT1     015414          OCTASC- 104402          RESREG  012510          TRPTAB  015136          T5      002270
DT2     015426          OCTASN  013226          RESTAR  014546          TWRD0   014310          T6      002446
EM1     015232          PARAM - 104405          RESTRT  012220          TWRD1   014312          T7      002624
EM2     015267          PARAMS  013042          RES05 - 104407          TWRD10  014330          VEC1    001164
EM3     015343          PARAM1  013072          RETRN   013600          TWRD11  014332          VEC2    001174
ENDCOD  015444          PARERR  013146          RS05    012654          TWRD12  014334          WRDCNT  013372
ENDFLG  013640          PASARG  012224          SAVPC   013630          TWRD13  014336          X     - 000000
EOP     012132          PASCNT  013574          SAVR0   013612          TWRD14  014340          XBIT  - 000000
ERRCNT  013576          PASTXT  015077          SAVR1   013614          TWRD15  014342          XLINE - 000020
ERRFLG  013572          PFAIL   014510          SAVR2   013616          TWRD16  014344          XN    - 000042
ERRMSG  012470          PFTAB   014650          SAVR3   013620          TWRD17  014346          Y     - 000011
ERRORS  012350

. ABS.  015446      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  19200 WORDS  ( 75 PAGES)
```

SYMBOL TABLE

DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
CZDHMC.BIN,CZDHMC.SEQ=CZDHMC.DOC,DHMACA.MAC,CZDHMC.P11