

DH11

DH11 CHA LEN BAS DAT
CZDHEC0

AH-FG24C-MC
1 OF 1 OCT 1985
COPYRIGHT© 1972-85

digital
MADE IN USA

The image displays a grid of 100 small, illegible data plots or charts arranged in 10 rows and 10 columns on a dark background. Each plot appears to contain some form of data visualization, possibly a bar chart or a small table, but the text and numbers are too small to be read. The plots are organized in a regular grid pattern, with some plots in the bottom right corner appearing to have a different layout or content than the others.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

.REM !

IDENTIFICATION

PRODUCT CODE: AC-8460C-MC
PRODUCT NAME: CZDHECO DH11 CHARACTER LENGTH/BASIC DATA TEST
DATE: JUNE 1985
MAINTAINER: NAC SOFTWARE ENGINEERING
AUTHOR: MICHAEL DAVIS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES

COPYRIGHT (C) 1985 BY DIGITAL EQUIPMENT CORPORATION

1
2
3
4
5
6
7
8
9

1. ABSTRACT

THE DH11 CHARACTER LENGTH AND BASIC DATA TEST VERIFIES THAT CHARACTER LENGTH CAN BE SELECTED CORRECTLY ON EACH LINE CORRECTLY, AND THAT THE CORRECT LINE NUMBER AND CHARACTER STATUS ARE RECEIVED ON EACH LINE SELECTED FOR TRANSMISSION.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

2. 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. LOADING PROCEDURE

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

4. 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 I 0002000

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 CHARACTER LENGTH AND BASIC DATA TEST" AND WILL THEN TYPE "VECTOR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

4.3 (CONT'D)

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 MESSAGE 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 CHARACTER LENGTH AND BASIC DATA TEST" AND WILL THEN CONTINUE AS DESCRIBED IN 4.3.1.9

4.3.3 PROGRAM 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 CHARACTER LENGTH AND BASIC DATA 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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

5. 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 IMMEDIATLY 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.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

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

- A) IF SW10=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST IN SEQUENCE, AFTER CLEARING ERROR FLAGS.
- B) IF SW11=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST SEQUENCE, AFTER CLEARING ERROR FLAGS.
- C) 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.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

5.2.7 ERRORS (ERROR HANDLER)

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

- A) 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.
- B) IF THE TEST THAT FAILED DID NOT FAIL PREVIOUSLY DURING THIS PASS, A COMPLETE ERROR REPORT IS MADE IF THE TEST THAT FAILED FAILED MOR THAT ONCE DURING THE CURRENT PASS, ONLY THE DATA RELATING TO THE FAILUER IS TYPED. IF SW13=1, NO ERROR TYPEOUT IS MADE.
- C) 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 RO. IF SW15=0, THE PROGRAM WILL NOT HALT, BUT WILL CHECK FOR ESCAPE TO NEXT TEST.
- D) 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 TH 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.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

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 "CZDHE-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 "CZDHE-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.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

5. (CONT'D)

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

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 PARAMETER WITHIN
A TEST, SET SW09=1 TO FREEZE THE DATA
(SEE LISTING FOR THOSE TESTS THAT INCORPORATE THIS FEATURE)

6. (CONT'D)

6.3.3 PROGRAM START TO SCOPE LOOP ON SELECTED TEST

PERFORM SECTION 4.3.4 WITH SW14=1

7. RESTRICTIONS

7.1 STARTING

THE DH11 TEST CARD MUST BE INSTALLED

7.2 RUNNING

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

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

PROCESSOR	TIME
PDP-11/05,10	
PDP-11/20	
PDP-11/40	
PDP-11/45	

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

9. PROGRAM DESCRIPTION

THIS PROGRAM CONSISTS OF 64 (DECIMAL) TESTS THAT CHECK, IN INDIVIDUAL TEST LOOPS, CHARACTER LENGTH SELECTION FOR EACH LINE AT EACH LENGTH OF 5,6,7, OR 8 BITS PER CHARACTER.

A CHARACTER CODE OF 377 IS TRANSMITTED ON A EACH LINE AT 5,6,7, AND 8 BITS PER CHARACTER. THE RECEIVED CHARACTER IS CHECKED TO VERIFY THAT THE DATA IS CORRECT (A CODE OF 37, 77, 177, OR 377 IF THE LENGTH IS 5,6,7, OR 8 BITS, RESPECTIVELY), AND THAT THE RECEIVED LINE NUMBER AND CHARACTER STATUS INFORMATION ARE CORRECT.

10. 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
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"
```

; 3

NUM
5 000000

```
.LIST ME
.NLIST MC,MD,CND
.HEADER †/1972,1985/,†/DH11 CHARACTER LENGTH AND BASIC DATA TEST/,†/CZDHE-CO/
```

```
;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE DH11 CHARACTER LENGTH AND BASIC DATA 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>
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE " CZDHE-CO "
;AND THEN RESUM TESTING
```

; 3

000000

```
.TITLE CZDHE-CO
.ENABLE ABS
.NLIST MC,MD,CND
.LIST ME
.SYMBOLS
```

6 000000

;SWITCH REGISTER OPTIONS

100000
040000
020000
010000
004000
002000
001000
000400
000100
000040
000020
000010
000004
000002
000001

```
SW15=100000      ;=1,HALT ON ERROR
SW14=40000       ;=1,LOOP ON CURRENT TEST
SW13=20000       ;=1,INHIBIT ERROR TYPEOUT
SW12=10000
SW11=4000        ;=1,INHIBIT ITERATIONS
SW10=2000        ;=1,ESCAPE TO NEXT TEST ON ERROR
SW09=1000        ;=1,LOOP WITH CURRENT DATA
SW08=400
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1
```

; 3

```
;RESTART PROGRAM AT SELECTED TEST
;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

```

;SWR=177570 ;CONSOLE SWITCH REGISTER ; 3
;LIGHTS=177570 ;PDP-11/45 DISPLAY REGISTER ; 4
177776      PS=177776 ;PROCESSOR STATUS WORD ; 4
017004      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      PUSHRO=10046 ;SAVE R0 ON STACK
012600      POPRO=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
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

```

; 3

0

```

000000 000000 ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
000200 .-0
      .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          000200          000167 000600          .-200      ;STANDARD INTERRUPT VECTORS
          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          104400          000001          TRPDEF  SCOPE,+/SCOPE LOOP AND ITERATION HANDLER/
          104401          000001          SCOPE=TRAP+Y          ;SCOPE LOOP AND ITERATION HANDLER
          000204          104401          000001          Y=Y+1

000204          104401          000002          TRPDEF  TYPE,+/TELETYPE OUTPUT ROUTINE/
          104401          000002          TYPE=TRAP+Y          ;TELETYPE OUTPUT ROUTINE
          000204          104401          000002          Y=Y+1

000204          104402          000003          TRPDEF  OCTASC,+/OCTAL TO ASCII CONVERSION/
          104402          000003          OCTASC=TRAP+Y        ;OCTAL TO ASCII CONVERSION
          000204          104402          000003          Y=Y+1

000204          104403          000004          TRPDEF  INSTR,+/INPUT ASCII STRING/
          104403          000004          INSTR=TRAP+Y         ;INPUT ASCII STRING
          000204          104403          000004          Y=Y+1

000204          104404          000005          TRPDEF  INSTER,+/STRING INPUT ERROR/
          104404          000005          INSTER=TRAP+Y       ;STRING INPUT ERROR
          000204          104404          000005          Y=Y+1

000204          104405          000006          TRPDEF  PARAM,+/CONVERT STRING TO OCTAL, CHECK LIMITS/
          104405          000006          PARAM=TRAP+Y        ;CONVERT STRING TO OCTAL, CHECK LIMITS
          000204          104405          000006          Y=Y+1

000204          104406          000007          TRPDEF  SAV05P,+/SAVE R0-R5, PC/
          104406          000007          SAV05P=TRAP+Y       ;SAVE R0-R5, PC
          000204          104406          000007          Y=Y+1

000204          104407          000010          TRPDEF  RES05,+/RESTORE R0-R5/
          104407          000010          RES05=TRAP+Y        ;RESTORE R0-R5
          000204          104407          000010          Y=Y+1

000204          104410          000011          TRPDEF  SCOPE1,+/CHECK FOR FREEZE ON CURRENT DATA/
          104410          000011          SCOPE1=TRAP+Y       ;CHECK FOR FREEZE ON CURRENT DATA
          000204          104410          000011          Y=Y+1

2          000046          014456          .-46
3          000046          014456          LOGICAL
4          000052          040000          .-52
5          000052          040000          40000
6          .MACRO      CODEM1
7          MOV        DHSSR,DHSLR          ;SET UP ADDRESS OF SILO
8          INC        DHSLR              ;STATUS REGISTER HIGH BYTE
9          .ENDM      CODEM1
10 000054          .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

001004 012767 000340 176764 START: MOV #340,PS ;LOCK OUT INTERRUPTS ; 4
001012 012706 017004 MOV #STACK,SP ;SET UP PROCESSOR STACK
001016 012702 000024 MOV #24,R2 ; POINT TO VECTOR AREA ; 7
001022 012722 015760 MOV #PFAIL,(R2)+ ;SET UP POWER FAIL TRAP ; 7
001026 012722 000340 MOV #340,(R2)+ ;SERVICE AT LEVEL 7 ; 7
001032 012722 014616 MOV #ERRORS,(R2)+ ;ERROR HANDLER ; 7
001036 012722 000340 MOV #340,(R2)+ ;SERVICE AT LEVEL 7 ; 7
001042 012722 015030 MOV #TRPSRV,(R2)+ ;GENERAL HANDLER DISPATCH SERVICE ; 7
001046 012712 000340 MOV #340,(R2) ;SERVICE AT LEVEL 7 ; 8
001052 005067 014674 CLR STFLG ;CLEAR TEST START FLAG
001056 005067 014630 CLR PASCNT ;CLEAR PASS COUNT
001062 005067 014626 CLR ERRCNT ;CLEAR ERROR COUNT
001066 005067 014616 CLR ERRFLG ;CLEAR ERROR FLAG
001072 005067 014612 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 016130 TYPE ,MTITLE ;TYPE TITLE MESSAGE ; 4
001154 005767 014570 TST INIFLG ;CHECK INITIALIZATION FLAG

.IF NB <DHRVEC> 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$,@#4
1$: TST (R0)+ ;SET UP TIME OUT RETURN ; 9
BR 1$ ;WILL TRAP WHEN NO MEMORY ;LOCATION RESPONDED, CONTINUE
2$: MOV R0,HCORE ;R0 CONTAINS ADDRESS OF ;NON EXISTANT MEMORY ; 9
SUB #2,HCORE
MOV #5,@#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 ;RESTORE TRAPCATCHER
    MOV @TRTRET, @#16 ;SET UP TRACE TRAP VECTOR

.ENDC
.IF NB <DHRVEC>
.IF B <> ; 3
        BR VEC2
.IFF
        TST INIFLG ;IF INITIALIZE FLAG=0
        BEQ VEC2 ;GET VECTOR AND CSR ADDRESS

.ENDC
VEC1: BIT #SW00, @SWR ;IF SW00=1, GET NEW VECTOR ; 4
      BEQ BEGIN ;AND CSR ; 4
VEC2: MOV #300, R1 ; 4
      MOV #302, R2 ; 4
      MOV #4, R3
1$: MOV R2, (R1) ;RESTORE TRAPCATCHER
    CLR (R2) ;IN FLOATING VECTOR AREA
    ADD R3, R1
    ADD R3, R2
    CMP R1, #1000
    BNE 1$

;INPUT ADDRESS OF DEVICE VECTOR
;MESSAGE "VECTOR ADDRESS-"
;CONVERT STRING TO OCTAL
;LOW LIMIT
;HIGH LIMIT ; 3
;LOCATIONS TO BE FILLED
;NUMBER OF LOCATIONS
;LSB MASK
;INPUT ADDRESS OF DEVICE CSR
;MESSAGE "CONTROL REGISTER ADDRESS-"
;CONVERT STRING TO OCTAL
;LOW LIMIT
;HIGH LIMIT
;LOCATIONS TO BE FILLED
;NUMBER OF LOCATIONS
;LSB MASK

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

```

001162	000404				
001164	032777	000001	177606		
001172	001445				
001174					
001174	012701	000300			
001200	012702	000302			
001204	012703	000004			
001210	010211				
001212	005012				
001214	060301				
001216	060302				
001220	020127	001000			
001224	001371				
001226	104403				
001230	016210				
001232	104405				
001234	000300				
001236	000770				
001240	015700				
001242	003				
001243	004				
001244	104403				
001246	016232				
001250	104405				
001252	000000				
001254	177776				
001256	015656				
001260	007				
001261	010				
001262					
001262	016767	014406	014406		
001270	005267	014402			
001274	005767	014450			
001300	001002				
001302	005167	014442			

```

001306 012767 000340 176462 BEGIN:  MOV    #340,PS           ;LOCK OUT INTERRUPTS
001314 012706 017004           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 016421           MTSTPC                   ;MESSAGE "TEST PC"
001334 104405           PARAM                    ;CONVERT STRING TO OCTAL
001336 000000           0
001340 017500           17500
001342 015716           RETRN
001344    001           .BYTE 1
001345    001           .BYTE 1
001346 000410           BR    2$
001350 012767 001400 014340 1$:  MOV    #T1,RETRN        ;NORMAL START, TEST 1
001356 005767 014370           TST    STFLG             ;IF LOOPING, BYPASS TYPEOUT
001362 001004           BNE    3$
001364 005167 014362           COM    STFLG
001370 104401 016415           TYPE  ,MR
001374 000177 014316           3$:  JMP    @RETRN          ;TYPE "R" TO INDICATE START ; 3
                                ;START TESTING

```

```

2      000020      XLINE=LINEX
3      000000      XBIT=BITX
4      000004      XLENGTH=LENGTX
5      000011      CLENGTH=LENGTC
6      000777      CODEX=XCXDE
7      107777      RDATA=DATAR
8      000000      LINEX=0
9      000001      BITX=1
11     000020      .REPT 20
12                                     .NLIST
13     LENGTHX=0
14     LENGTC=5
15     XCXDE=37
16     .LIST
17     .REPT 4
18     .NLIST
19     DATAR=LINEX*400+100000*XCXDE
20     .LIST
21     CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCXDE,\DATAR
22     .NLIST
23     LENGTX=LENGTX+1
24     LENGTC=LENGTC+1
25     XCXDE=XCXDE*2+1
26     .LIST
27     .ENDR
28     .NLIST
29     LINEX=LINEX+1
30     BITX=BITX*BITX
31     .LIST
32     .ENDR

```

```

000000
000005
000037
000004

```

```

LENGTX=0
LENGTC=5
XCXDE=37
.REPT 4
.NLIST
DATAR=LINEX*400+100000*XCXDE
.LIST
CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCXDE,\DATAR
.NLIST
LENGTX=LENGTX+1
LENGTC=LENGTC+1
XCXDE=XCXDE*2+1
.LIST
.ENDR
DATAR=LINEX*400+100000*XCXDE
CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCXDE,\DATAR

```

```

001400 100037

```

```

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 0
;CHARACTER LENGTH IS 5 BITS
;EXPECTED RECEIVED CHARACTER IS 37
;LINE SPEED IS 9600 BAUD

```

```

001400      TS \XN,400,2#
001400 012767 000340 176370 T1:  MOV  #340,PS ;DISABLE ALL INTERRUPTS
001406 012767 000400 014310      MOV  #400,ICOUNT ;SET UP FOR 400 ITERATIONS
001414 012767 001526 014276      MOV  #2#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

```

```

                .IF NB <>
                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA      ; 3
                .ENDC
                XN=XN+1
001422 000002      MOV     #BIT11,@DHSCR        ;MASTER CLEAR INTERFACE
001430 012777 004000 014226      MOV     #37,TDATA        ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
001436 012777 000000 014212      MOV     #0,@DHSCR        ;SELECT LINE 0
001444 012777 177777 014214      MOV     #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
001452 012777 015756 014204      MOV     @TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
001460 012777 033500 014174      MOV     #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
001466 052777 000000 014166      BIS     #0,@DHLPR        ;SET CHARACTER LENGTH FOR 5 BITS
001474 012777 000001 014166      MOV     #1,@DHBAR        ;START TRANSMITTER
001502 105777 014150      1#:   TSTB     @DHSCR        ;WAIT TO RECEIVE CHARACTER
001506 100375      BPL     1#
001510 017704 014144      MOV     @DHNRC,R4        ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
001514 012705 100037      MOV     #100037,R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
001520 020504      CMP     R5,R4          ;STATUS IN HIGH BYTE
001522 001401      BEQ     2#            ;ARE EXPECTED AND RECEIVED DATA THE SAME
001524      HLT
001524 104000      EMT                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
001526 104400      2#:   SCOPE
                                LENGTHX=LENGTHX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                DATAR=LINEX+400+100000+XCODE
001530      CHART1 \LINEX,\BITX,\LENGTHX,\LENGTC,\XCODE,\DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 0
                                ;CHARACTER LENGTH IS 6 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 77
                                ;LINE SPEED IS 9600 BAUD
001530      TS \XN,400,2#
001530 012767 000340 176240      T2:   MOV     #340,PS        ;DISABLE ALL INTERRUPTS
001536 012767 000400 014160      MOV     #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
001544 012767 001656 014146      MOV     #2#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB <>
                MOV     #,FREEZ1                ;SET UP TO LOOP WITH DATA      ; 3
                .ENDC
                XN=XN+1
001552 000003      MOV     #BIT11,@DHSCR        ;MASTER CLEAR INTERFACE
001560 012767 000077 014170      MOV     #77,TDATA        ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
001566 012777 000000 014062      MOV     #0,@DHSCR        ;SELECT LINE 0
001574 012777 177777 014064      MOV     #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
001602 012777 015756 014054      MOV     @TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
001610 012777 033500 014044      MOV     #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
001616 052777 000001 014036      BIS     #1,@DHLPR        ;SET CHARACTER LENGTH FOR 6 BITS
001624 012777 000001 014036      MOV     #1,@DHBAR        ;START TRANSMITTER
001632 105777 014020      1#:   TSTB     @DHSCR        ;WAIT TO RECEIVE CHARACTER
001636 100375      BPL     1#
001640 017704 014014      MOV     @DHNRC,R4        ;(R4)=RECEIVED CHARACTER

```

```

001644 012705 100077          MOV      #100077,R5          ;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
; (R5)-EXPECTED CHARACTER IN LOW BYTE
; AND LINE NUMBER AND CHARACTER
; STATUS IN HIGH BYTE
; ARE EXPECTED AND RECEIVED DATA THE SAME

001650 020504          CMP      R5,R4
001652 001401          BEQ     2#
001654          HLT
001654 104000          EMT          ;CHARACTER LENGTH, DATA
;OR LINE NUMBER ERROR

001656 104400          2#:      SCOPE
000002          LENGTX=LENGTX+1
000007          LENGTC=LENGTC+1
000177          XCODE=XCODE+2+1
001660 100177          DATAR=LINEX+400+100000*XCODE
          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 0
;CHARACTER LENGTH IS 7 BITS
;EXPECTED RECEIVED CHARACTER IS 177
;LINE SPEED IS 9600 BAUD

001660          TS \XN,400,2#
001660 012767 000340 176110  T3:      MOV      #340,PS          ;DISABLE ALL INTERRUPTS
001666 012767 000400 014030      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
001674 012767 002006 014016      MOV      #2#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

          .IF NB <>
          MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1

001702 012777 004000 013746      MOV      #BIT11,SDHSCR      ;MASTER CLEAR INTERFACE
001710 012767 000177 014040      MOV      #177,TDATA        ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
001716 012777 000000 013732      MOV      #0,SDHSCR         ;SELECT LINE 0
001724 012777 177777 013734      MOV      #-1,SDHBC         ;SET UP TO TRANSMIT 1 BYTE
001732 012777 015756 013724      MOV      #TDATA,SDHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
001740 012777 033500 013714      MOV      #33500,SDHLPR     ;SET LINE SPEED FOR 9600 BAUD
001746 052777 000002 013706      BIS      #2,SDHLPR         ;SET CHARACTER LENGTH FOR 7 BITS
001754 012777 000001 013706      MOV      #1,SDHBR         ;START TRANSMITTER
001762 105777 013670          1#:      TSTB   SDHSCR         ;WAIT TO RECEIVE CHARACTER
001766 100375          BPL     1#
001770 017704 013664          MOV      #DHNRC,R4

; (R4)-RECEIVED CHARACTER
; IN LOW BYTE, AND LINE NUMBER AND
; CHARACTER STATUS IN HIGH BYTE
; (R5)-EXPECTED CHARACTER IN LOW BYTE
; AND LINE NUMBER AND CHARACTER
; STATUS IN HIGH BYTE
; ARE EXPECTED AND RECEIVED DATA THE SAME

001774 012705 100177          MOV      #100177,R5

002000 020504          CMP      R5,R4
002002 001401          BEQ     2#
002004          HLT
002004 104000          EMT          ;CHARACTER LENGTH, DATA
;OR LINE NUMBER ERROR

002006 104400          2#:      SCOPE
000003          LENGTX=LENGTX+1
000010          LENGTC=LENGTC+1
000377          XCODE=XCODE+2+1
100377          DATAR=LINEX+400+100000*XCODE

```

002010

CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

```

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 0
;CHARACTER LENGTH IS 10 BITS
;EXPECTED RECEIVED CHARACTER IS 377
;LINE SPEED IS 9600 BAUD

```

```

002010 000005
002010 012767 000340 175760 TS \XN,400,2#
002016 012767 000400 013700 T4:  MOV  #340,PS ;DISABLE ALL INTERRUPTS
002024 012767 002136 013666  MOV  #400,ICOUNT ;SET UP FOR 400 ITERATIONS
  MOV  #2#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
  .IF NB <>
  MOV  #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
  .ENDC
  XN=XN+1
002032 012777 004000 013616  MOV  #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
002040 012767 000377 013710  MOV  #377,TDATA ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
002046 012777 000000 013602  MOV  #0,@DHSCR ;SELECT LINE 0
002054 012777 177777 013604  MOV  #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
002062 012777 015756 013574  MOV  #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002070 012777 033500 013564  MOV  #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
002076 052777 000003 013556  BIS  #3,@DHLPR ;SET CHARACTER LENGTH FOR 10 BITS
0C2104 012777 000001 013556  MOV  #1,@DHBAR ;START TRANSMITTER
002112 105777 013540 1# : TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
002116 100375  BPL 1#
002120 017704 013534  MOV  @DHNRC,R4 ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
002124 012705 100377  MOV  #100377,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
;AND LINE NUMBER AND CHARACTER
;STATUS IN HIGH BYTE
002130 020504  CMP  R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME
002132 001401  BEQ 2#
002134  HLT
002134 104000  EMT ;CHARACTER LENGTH, DATA
;OR LINE NUMBER ERROR
002136 104400 2# : SCOPE
  000004  LENGTX=LENGTX+1
  000011  LENGTC=LENGTC+1
  000777  XCODE=XCODE+2+1
  000001  LINEX=LINEX+1
  000002  BITX=BITX+BITX
  000000  LENGTX=0
  000005  LENGTC=5
  000037  XCODE=37
  .REPT 4
  .NLIST
  DATAR=LINEX+400+100000+XCODE
  .LIST
  CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
  .NLIST
  LENGTX=LENGTX+1
  LENGTC=LENGTC+1
  XCODE=XCODE+2+1
  .LIST
  .ENDR

```

```

002140 100437 DATAR=LINEX*400+100000+XCODE
          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
          ;CHARACTER LENGTH TEST
          ;TRANSMIT 1 CHARACTER ON LINE 1
          ;CHARACTER LENGTH IS 5 BITS
          ;EXPECTED RECEIVED CHARACTER IS 37
          ;LINE SPEED IS 9600 BAUD

002140 012767 000340 175630 TS \XN,400,2$
002146 012767 000400 013550 T5:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
002154 012767 002266 013536      MOV  #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
          MOV  #2$,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <>
          MOV  #,FREEZ1     ;SET UP TO LOOP WITH DATA           ; 3
          .ENDC
          XN=XN+1

002162 012777 004000 013466      MOV  #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
002170 012767 000037 013560      MOV  #37,TDATA         ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
002176 012777 000001 013452      MOV  #1,@DHSCR        ;SELECT LINE 1
002204 012777 177777 013454      MOV  #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
002212 012777 015756 013444      MOV  #TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002220 012777 033500 013434      MOV  #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
002226 052777 000000 013426      BIS  #0,@DHLPR        ;SET CHARACTER LENGTH FOR 5 BITS
002234 012777 000002 013426      MOV  #2,@DHBAR        ;START TRANSMITTER
002242 105777 013410 1$:  TSTB  @DHSCR          ;WAIT TO RECEIVE CHARACTER
002246 100375
002250 017704 013404      MOV  @DHNRC,R4        ;(R4)=RECEIVED CHARACTER
          ;IN LOW BYTE, AND LINE NUMBER AND
          ;CHARACTER STATUS IN HIGH BYTE
002254 012705 100437      MOV  #100437,R5       ;(R5)=EXPECTED CHARACTER IN LOW BYTE
          ;AND LINE NUMBER AND CHARACTER
          ;STATUS IN HIGH BYTE
          ;ARE EXPECTED AND RECEIVED DATA THE SAME

002260 020504      CMP  R5,R4
002262 001401      BEQ  2$
002264      HLT
002264 104000      EMT
          ;CHARACTER LENGTH, DATA
          ;OR LINE NUMBER ERROR

002266 104400 2$:  SCOPE
          000001  LENGTX=LENGTX+1
          000006  LENGTC=LENGTC+1
          000077  XCODE=XCODE+2+1
          100477  DATAR=LINEX*400+100000+XCODE
002270      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
          ;CHARACTER LENGTH TEST
          ;TRANSMIT 1 CHARACTER ON LINE 1
          ;CHARACTER LENGTH IS 6 BITS
          ;EXPECTED RECEIVED CHARACTER IS 77
          ;LINE SPEED IS 9600 BAUD

002270 012767 000340 175500 TS \XN,400,2$
002276 012767 000400 013420 T6:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
002304 012767 002416 013406      MOV  #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
          MOV  #2$,ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <>
          MOV  #,FREEZ1     ;SET UP TO LOOP WITH DATA           ; 3

```



```

002534 012705 100577          MOV      #100577,R5          ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                           ;AND LINE NUMBER AND CHARACTER
002540 020504          CMP      R5,R4          ;STATUS IN HIGH BYTE
002542 001401          BEQ      2$            ;ARE EXPECTED AND RECEIVED DATA THE SAME
002544          HLT
002544 104000          EMT          ;CHARACTER LENGTH, DATA
                                           ;OR LINE NUMBER ERROR

002546 104400          2$:      SCOPE
000003          LENGTX=LENGTX+1
000010          LENGTC=LENGTC+1
000377          XCODE=XCODE+2+1
100777          DATAR=LINEX+400+100000+XCODE
002550          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                           ;CHARACTER LENGTH TEST
                                           ;TRANSMIT 1 CHARACTER ON LINE 1
                                           ;CHARACTER LENGTH IS 10 BITS
                                           ;EXPECTED RECEIVED CHARACTER IS 377
                                           ;LINE SPEED IS 9600 BAUD

002550          TS \XN,400,2$
002550 012767 000340 175220 T10:    MOV      #340,PS          ;DISABLE ALL INTERRUPTS
002556 012767 000400 013140      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
002564 012767 002676 013126      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                           .IF NB <>
                                           MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                           .ENDC
                                           XN=XN+1

002572 000011          000011
002572 012777 004000 013056      MOV      #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
002600 012767 000377 013150      MOV      #377,TDATA      ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
002606 012777 000001 013042      MOV      #1,@DHSCR       ;SELECT LINE 1
002614 012777 177777 013044      MOV      #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
002622 012777 015756 013034      MOV      #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002630 012777 033500 013024      MOV      #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
002636 052777 000003 013016      BIS      #3,@DHLPR       ;SET CHARACTER LENGTH FOR 10 BITS
002644 012777 000002 013016      MOV      #2,@DHBAR       ;START TRANSMITTER
002652 105777 013000          1$:    TSTB    @DHSCR       ;WAIT TO RECEIVE CHARACTER
002656 100375          BPL      1$
002660 017704 012774          MOV      @DHNRC,R4

                                           ;(R4)=RECEIVED CHARACTER
                                           ;IN LOW BYTE, AND LINE NUMBER AND
                                           ;CHARACTER STATUS IN HIGH BYTE
002664 012705 100777          MOV      #100777,R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                           ;AND LINE NUMBER AND CHARACTER
                                           ;STATUS IN HIGH BYTE
                                           ;ARE EXPECTED AND RECEIVED DATA THE SAME

002670 020504          CMP      R5,R4
002672 001401          BEQ      2$
002674          HLT          ;CHARACTER LENGTH, DATA
002674 104000          EMT          ;OR LINE NUMBER ERROR

002676 104400          2$:    SCOPE
000004          LENGTX=LENGTX+1
000011          LENGTC=LENGTC+1
000777          XCODE=XCODE+2+1
000002          LINEX=LINEX+1
000004          BITX=BITX+BITX
000000          LENGTX=0

```

```

000005          LENGTC=5
000037          XCODE=37
000004          .REPT 4
                .NLIST
                DATAR=LINEX*400+100000+XCODE
                .LIST
                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                .NLIST
                LENGTX=LENGTX+1
                LENGTC=LENGTC+1
                XCODE=XCODE+2+1
                .LIST
                .ENDR
002700 101037   DATAR=LINEX*400+100000+XCODE
                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                ;CHARACTER LENGTH TEST
                ;TRANSMIT 1 CHARACTER ON LINE 2
                ;CHARACTER LENGTH IS 5 BITS
                ;EXPECTED RECEIVED CHARACTER IS 37
                ;LINE SPEED IS 9600 BAUD

002700          TS \XN,400,2$
002700 012767 000340 175070 T11:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
002706 012767 000400 013010      MOV    #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
002714 012767 003026 012776      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

                .IF NB <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1

002722 012777 004000 012726      MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
002730 012767 000037 013020      MOV    #37,TDATA       ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
002736 012777 000002 012712      MOV    #2,@DHSCR       ;SELECT LINE 2
002744 012777 177777 012714      MOV    #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
002752 012777 015756 012704      MOV    #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
002760 012777 033500 012674      MOV    #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
002766 052777 000000 012666      BIS    #0,@DHLPR       ;SET CHARACTER LENGTH FOR 5 BITS
002774 012777 000004 012666      MOV    #4,@DHBAR       ;START TRANSMITTER
003002 105777 012650          1$:  TSTB   @DHSCR         ;WAIT TO RECEIVE CHARACTER
003006 100375          BPL    1$
003010 017704 012644          MOV    @DHNRC,R4

                ;(R4)=RECEIVED CHARACTER
                ;IN LOW BYTE, AND LINE NUMBER AND
                ;CHARACTER STATUS IN HIGH BYTE
                ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                ;AND LINE NUMBER AND CHARACTER
                ;STATUS IN HIGH BYTE
                ;ARE EXPECTED AND RECEIVED DATA THE SAME

003014 012705 101037          MOV    #101037,R5

                ;CHARACTER LENGTH, DATA
                ;OR LINE NUMBER ERROR

003020 020504          CMP    R5,R4
003022 001401          BEQ   2$
003024          HLT
003024 104000          EMT

003026 104400          2$:  SCOPE
                000001  LENGTX=LENGTX+1
                000006  LENGTC=LENGTC+1
                000077  XCODE=XCODE+2+1
                101077  DATAR=LINEX*400+100000+XCODE
003030          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

```

```

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 2
;CHARACTER LENGTH IS 6 BITS
;EXPECTED RECEIVED CHARACTER IS 77
;LINE SPEED IS 9600 BAUD

003030      003030      012767      000340      174740      TS \XN,400,2#
003036      012767      000400      012660      T12:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
003044      012767      003156      012646      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
                                MOV      #2#,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1           ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1

003052      000013      003052      012777      004000      012576      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
003060      012777      000077      012670      MOV      #77,TDATA          ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
003066      012777      000002      012562      MOV      #2,@DHSCR          ;SELECT LINE 2
003074      012777      177777      012564      MOV      #-1,@DHBC          ;SET UP TO TRANSMIT 1 BYTE
003102      012777      015756      012554      MOV      #TDATA,@DHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003110      012777      033500      012544      MOV      #33500,@DHLPR      ;SET LINE SPEED FOR 9600 BAUD
003116      052777      000001      012536      BIS      #1,@DHLPR          ;SET CHARACTER LENGTH FOR 6 BITS
003124      012777      000004      012536      MOV      #4,@DHBR          ;START TRANSMITTER
003132      105777      012520      1#:  TSTB      @DHSCR          ;WAIT TO RECEIVE CHARACTER
003136      100375      BPL      1#
003140      017704      012514      MOV      @DHNRC,R4          ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
003144      012705      101077      MOV      #101077,R5        ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
003150      020504      CMP      R5,R4              ;ARE EXPECTED AND RECEIVED DATA THE SAME
003152      001401      BEQ      2#
003154      HLT
003154      104000      EMT                          ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

003156      104400      2#:  SCOPE
                                000002      LENGTX=LENGTX+1
                                000007      LENGTC=LENGTC+1
                                000177      XCODE=XCODE+2+1
                                101177      DATAR=LINEX*400+100000+XCODE
003160      003160      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 2
;CHARACTER LENGTH IS 7 BITS
;EXPECTED RECEIVED CHARACTER IS 177
;LINE SPEED IS 9600 BAUD

003160      003160      012767      000340      174610      TS \XN,400,2#
003166      012767      000400      012530      T13:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
003174      012767      003306      012516      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
                                MOV      #2#,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1           ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1

                                000014

```

```

003202 012777 004000 012446      MOV      #BIT11, @DHSCR      ;MASTER CLEAR INTERFACE
003210 012767 000177 012540      MOV      #177, TDATA        ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
003216 012777 000002 012432      MOV      #2, @DHSCR         ;SELECT LINE 2
003224 012777 177777 012434      MOV      #-1, @DHBC         ;SET UP TO TRANSMIT 1 BYTE
003232 012777 015756 012424      MOV      @TDATA, @DHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003240 012777 033500 012414      MOV      #33500, @DHLPR     ;SET LINE SPEED FOR 9600 BAUD
003246 052777 000002 012406      BIS      #2, @DHLPR         ;SET CHARACTER LENGTH FOR 7 BITS
003254 012777 000004 012406      MOV      #4, @DHBAR         ;START TRANSMITTER
003262 105777 012370      1$:    TSTB      @DHSCR        ;WAIT TO RECEIVE CHARACTER
003266 100375                      BPL      1$
003270 017704 012364      MOV      @DHNRC, R4

; (R4)=RECEIVED CHARACTER
; IN LOW BYTE, AND LINE NUMBER AND
; CHARACTER STATUS IN HIGH BYTE
003274 012705 101177      MOV      #101177, R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
; AND LINE NUMBER AND CHARACTER
; STATUS IN HIGH BYTE
; ARE EXPECTED AND RECEIVED DATA THE SAME

003300 020504                      CMP      R5, R4
003302 001401                      BEQ      2$
003304                      HLT
003304 104000                      EMT

; CHARACTER LENGTH, DATA
; OR LINE NUMBER ERROR

003306 104400      2$:    SCOPE
000003      LENGTX=LENGTX+1
000010      LENGTC=LENGTC+1
000377      XCODE=XCODE+2+1
101377      DATAR=LINEX*400+100000+XCODE
003310      CHART1 \LINEX, \BITX, \LENGTX, \LENGTC, \XCODE, \DATAR

; CHARACTER LENGTH TEST
; TRANSMIT 1 CHARACTER ON LINE 2
; CHARACTER LENGTH IS 10 BITS
; EXPECTED RECEIVED CHARACTER IS 377
; LINE SPEED IS 9600 BAUD

003310      TS \XN, 400, 2$
003310 012767 000340 174460      T14:    MOV      #340, PS      ;DISABLE ALL INTERRUPTS
003316 012767 000400 012400      MOV      #400, ICOUNT      ;SET UP FOR 400 ITERATIONS
003324 012767 003436 012366      MOV      #2$, ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

; IF NB <>
MOV      #, FREEZ1      ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1

003332 012777 004000 012316      MOV      #BIT11, @DHSCR     ;MASTER CLEAR INTERFACE
003340 012767 000377 012410      MOV      #377, TDATA        ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
003346 012777 000002 012302      MOV      #2, @DHSCR         ;SELECT LINE 2
003354 012777 177777 012304      MOV      #-1, @DHBC         ;SET UP TO TRANSMIT 1 BYTE
003362 012777 015756 012274      MOV      @TDATA, @DHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003370 012777 033500 012264      MOV      #33500, @DHLPR     ;SET LINE SPEED FOR 9600 BAUD
003376 052777 000003 012256      BIS      #3, @DHLPR         ;SET CHARACTER LENGTH FOR 10 BITS
003404 012777 000004 012256      MOV      #4, @DHBAR         ;START TRANSMITTER
003412 105777 012240      1$:    TSTB      @DHSCR        ;WAIT TO RECEIVE CHARACTER
003416 100375                      BPL      1$
003420 017704 012234      MOV      @DHNRC, R4

; (R4)=RECEIVED CHARACTER
; IN LOW BYTE, AND LINE NUMBER AND
; CHARACTER STATUS IN HIGH BYTE
003424 012705 101377      MOV      #101377, R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
; AND LINE NUMBER AND CHARACTER

```

```

003430 020504          CMP      R5,R4          ;STATUS IN HIGH BYTE
003432 001401          BEQ      2$              ;ARE EXPECTED AND RECEIVED DATA THE SAME
003434          HLT
003434 104000          EMT              ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

003436 104400          2$:      SCOPE
000004          LENGTX=LENGTX+1
000011          LENGTC=LENGTC+1
000777          XCODE=XCODE+2+1
000003          LINEX=LINEX+1
000010          BITX=BITX+BITX
000000          LENGTX=0
000005          LENGTC=5
000037          XCODE=37
000004          .REPT 4
                                .NLIST
                                DATAR=LINEX+400+100000+XCODE
                                .LIST
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                .NLIST
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                .LIST
                                .ENDR
                                DATAR=LINEX+400+100000+XCODE
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

003440 101437          ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 3
                                ;CHARACTER LENGTH IS 5 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 37
                                ;LINE SPEED IS 9600 BAUD

003440          TS \XN,400,2$
003440 012767 000340 174330 T15:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
003446 012767 000400 012250      MOV      #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
003454 012767 003566 012236      MOV      #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1

003462 012777 004000 012166      MOV      #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
003470 012767 000037 012260      MOV      #37,TDATA       ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
003476 012777 000003 012152      MOV      #3,@DHSCR       ;SELECT LINE 3
003504 012777 177777 012154      MOV      #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
003512 012777 015756 012144      MOV      #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003520 012777 033500 012134      MOV      #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
003526 052777 000000 012126      BIS      #0,@DHLPR       ;SET CHARACTER LENGTH FOR 5 BITS
003534 012777 000010 012126      MOV      #10,@DHBAR      ;START TRANSMITTER
003542 105777 012110          1$:  TSTB     @DHSCR        ;WAIT TO RECEIVE CHARACTER
003546 100375          BPL      1$
003550 017704 012104          MOV      @DHNRC,R4       ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
                                ;(R5)=EXPECTED CHARACTER IN LOW BYTE

003554 012705 101437          MOV      #101437,R5

```

```

003560 020504          CMP      R5,R4          ;AND LINE NUMBER AND CHARACTER
003562 001401          BEQ      2$              ;STATUS IN HIGH BYTE
003564          HLT                               ;ARE EXPECTED AND RECEIVED DATA THE SAME
003564 104000          EMT                               ;CHARACTER LENGTH, DATA

003566 104400          2$:  SCOPE                          ;OR LINE NUMBER ERROR
      000001          LENGTX=LENGTX+1
      000006          LENGTC=LENGTC+1
      000077          XCODE=XCODE+2+1
      101477          DATAR=LINEX*400+100000+XCODE
003570          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

      ;CHARACTER LENGTH TEST
      ;TRANSMIT 1 CHARACTER ON LINE 3
      ;CHARACTER LENGTH IS 6 BITS
      ;EXPECTED RECEIVED CHARACTER IS 77
      ;LINE SPEED IS 9600 BAUD

003570          TS \XN,400,2$
003570 012767 000340 174200 T16:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
003576 012767 000400 012120      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
003604 012767 003716 012106      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
      .ENDC
      XN=XN+1

003612 000017          MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
003620 012777 004000 012036      MOV      #77,TDATA        ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
003626 012777 000077 012130      MOV      #3,@DHSCR        ;SELECT LINE 3
003634 012777 177777 012022      MOV      #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
003642 012777 015756 012024      MOV      @TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
003650 012777 033500 012004      MOV      #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
003656 052777 000001 011776      BIS      #1,@DHLPR        ;SET CHARACTER LENGTH FOR 6 BITS
003664 012777 000010 011776      MOV      #10,@DHBAR       ;START TRANSMITTER
003672 105777 011760      1$:  TSTB     @DHSCR        ;WAIT TO RECEIVE CHARACTER
003676 100375          BPL      1$
003700 017704 011754          MOV      @DHNRC,R4        ;(R4)=RECEIVED CHARACTER
      ;IN LOW BYTE, AND LINE NUMBER AND
      ;CHARACTER STATUS IN HIGH BYTE
      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
      ;AND LINE NUMBER AND CHARACTER
      ;STATUS IN HIGH BYTE
      ;ARE EXPECTED AND RECEIVED DATA THE SAME

003704 012705 101477          MOV      #101477,R5

003710 020504          CMP      R5,R4          ;CHARACTER LENGTH, DATA
003712 001401          BEQ      2$              ;OR LINE NUMBER ERROR
003714          HLT
003714 104000          EMT

003716 104400          2$:  SCOPE
      000002          LENGTX=LENGTX+1
      000007          LENGTC=LENGTC+1
      000177          XCODE=XCODE+2+1
      101577          DATAR=LINEX*400+100000+XCODE
003720          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

      ;CHARACTER LENGTH TEST

```

```

;TRANSMIT 1 CHARACTER ON LINE 3
;CHARACTER LENGTH IS 7 BITS
;EXPECTED RECEIVED CHARACTER IS 177
;LINE SPEED IS 9600 BAUD

```

```

003720      000020      TS \XN,400,2#
003720 012767 000340 174050 T17:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
003726 012767 000400 011770      MOV    #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
003734 012767 004046 011756      MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                        MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                        .ENDC
                        XN=XN+1
003742 012777 004000 011706      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
003750 012767 000177 012000      MOV    #177,TDATA       ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
003756 012777 000003 011672      MOV    #3,@DHSCR        ;SELECT LINE 3
003764 012777 177777 011674      MOV    #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
003772 012777 015756 011664      MOV    #TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004000 012777 033500 011654      MOV    #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
004006 052777 000002 011646      BIS    #2,@DHLPR        ;SET CHARACTER LENGTH FOR 7 BITS
004014 012777 000010 011646      MOV    #10,@DHBAR       ;START TRANSMITTER
004022 105777 011630      1#:  TSTB   @DHSCR        ;WAIT TO RECEIVE CHARACTER
004026 100375      BPL    1#
004030 017704 011624      MOV    @DHNRC,R4        ;(R4)=RECEIVED CHARACTER
                        ;IN LOW BYTE, AND LINE NUMBER AND
                        ;CHARACTER STATUS IN HIGH BYTE
004034 012705 101577      MOV    #101577,R5       ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                        ;AND LINE NUMBER AND CHARACTER
004040 020504      CMP    R5,R4            ;STATUS IN HIGH BYTE
004042 001401      BEQ    2#              ;ARE EXPECTED AND RECEIVED DATA THE SAME
004044      HLT
004044 104000      EMT                    ;CHARACTER LENGTH, DATA
                        ;OR LINE NUMBER ERROR
004046 104400      2#:  SCOPE
000003      LENGTX=LENGTX+1
000010      LENGTC=LENGTC+1
000377      XCODE=XCODE+2+1
004050 101777      DATAR=LINEX+400+100000+XCODE
                        CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                        ;CHARACTER LENGTH TEST
                        ;TRANSMIT 1 CHARACTER ON LINE 3
                        ;CHARACTER LENGTH IS 10 BITS
                        ;EXPECTED RECEIVED CHARACTER IS 377
                        ;LINE SPEED IS 9600 BAUD
004050      TS \XN,400,2#
004050 012767 000340 173720 T20:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
004056 012767 000400 011640      MOV    #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
004064 012767 004176 011626      MOV    #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                        .IF NB  <>
                        MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                        .ENDC
                        XN=XN+1
004072 012777 004000 011556      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
004100 012767 000377 011650      MOV    #377,TDATA       ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)

```

```

004106 012777 000003 011542      MOV      #3, @DHSCR      ;SELECT LINE 3
004114 012777 177777 011544      MOV      #-1, @DHBC     ;SET UP TO TRANSMIT 1 BYTE
004122 012777 015756 011534      MOV      @TDATA, @DHBA  ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004130 012777 033500 011524      MOV      #33500, @DHLPR ;SET LINE SPEED FOR 9600 BAUD
004136 052777 000003 011516      BIS      #3, @DHLPR     ;SET CHARACTER LENGTH FOR 10 BITS
004144 012777 000010 011516      MOV      #10, @DHBAR    ;START TRANSMITTER
004152 105777 011500          1$: TSTB   @DHSCR          ;WAIT TO RECEIVE CHARACTER
004156 100375          BPL     1$
004160 017704 011474          MOV      @DHNRC, R4

                                ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
004164 012705 101777          MOV      #101777, R5    ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME

004170 020504          CMP      R5, R4
004172 001401          BEQ     2$
004174          HLT
004174 104000          EMT      ;CHARACTER LENGTH, DATA

004176 104400          2$: SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                LINEX=LINEX+1
                                BITX=BITX+BITX
                                LENGTX=0
                                LENGTC=5
                                XCODE=37
                                .REPT 4
                                .NLIST
                                DATAR=LINEX+400+100000+XCODE
                                .LIST
                                CHART1 \LINEX, \BITX, \LENGTX, \LENGTC, \XCODE, \DATAR
                                .NLIST
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                .LIST
                                .ENDR
                                DATAR=LINEX+400+100000+XCODE
                                CHART1 \LINEX, \BITX, \LENGTX, \LENGTC, \XCODE, \DATAR

                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 4
                                ;CHARACTER LENGTH IS 5 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 37
                                ;LINE SPEED IS 9600 BAUD

004200 102037          TS \XN, 400, 2$
004200 012767 000340 173570      T21: MOV      #340, PS    ;DISABLE ALL INTERRUPTS
004206 012767 000400 011510      MOV      #400, ICOUNT  ;SET UP FOR 400 ITERATIONS
004214 012767 004326 011476      MOV      #2$, ESCAPE   ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB <>
                                MOV      @, FREEZ1      ;SET UP TO LOOP WITH DATA
                                .ENDC
                                XN=XN+1
004222 012777 004000 011426      MOV      #BIT11, @DHSCR ;MASTER CLEAR INTERFACE
                                ; 3

```

```

004230 012767 000037 011520      MOV      #37,TDATA      ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
004236 012777 000004 011412      MOV      #4,@DHSCR     ;SELECT LINE 4
004244 012777 177777 011414      MOV      #-1,@DHBC     ;SET UP TO TRANSMIT 1 BYTE
004252 012777 015756 011404      MOV      @TDATA,@DHBA  ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004260 012777 033500 011374      MOV      #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
004266 052777 000000 011366      BIS      #0,@DHLPR     ;SET CHARACTER LENGTH FOR 5 BITS
004274 012777 000020 011366      MOV      #20,@DHBAR    ;START TRANSMITTER
004302 105777 011350      1$: TSTB    @DHSCR       ;WAIT TO RECEIVE CHARACTER
004306 100375      BPL      1$
004310 017704 011344      MOV      @DHNRC,R4

                                ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
004314 012705 102037      MOV      #102037,R5    ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME

004320 020504      CMP      R5,R4
004322 001401      BEQ     2$
004324      HLT
004324 104000      EMT
                                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

004326 104400      2$: SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                DATAR=LINEX+400+100000*XCODE
004330      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 4
                                ;CHARACTER LENGTH IS 6 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 77
                                ;LINE SPEED IS 9600 BAUD

004330      TS \XN,400,2$
004330 012767 000340 173440 T22: MOV      #340,PS      ;DISABLE ALL INTERRUPTS
004336 012767 000400 011360      MOV      #400,ICOUNT   ;SET UP FOR 400 ITERATIONS
004344 012767 004456 011346      MOV      #2$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      @,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1

004352 000023      MOV      @BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
004360 012777 004000 011276      MOV      #77,TDATA     ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
004366 012777 000004 011370      MOV      #4,@DHSCR     ;SELECT LINE 4
004374 012777 177777 011262      MOV      #-1,@DHBC     ;SET UP TO TRANSMIT 1 BYTE
004402 012777 015756 011254      MOV      @TDATA,@DHBA  ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004410 012777 033500 011244      MOV      #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
004416 052777 000001 011236      BIS      #1,@DHLPR     ;SET CHARACTER LENGTH FOR 6 BITS
004424 012777 000020 011236      MOV      #20,@DHBAR    ;START TRANSMITTER
004432 105777 011220      1$: TSTB    @DHSCR       ;WAIT TO RECEIVE CHARACTER
004436 100375      BPL      1$
004440 017704 011214      MOV      @DHNRC,R4

                                ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
004444 012705 102077      MOV      #102077,R5    ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE

```



```

;EXPECTED RECEIVED CHARACTER IS 377
;LINE SPEED IS 9600 BAUD

```

```

004610 000025 TS \XN,400,2#
004610 012767 000340 173160 T24: MOV #340,PS ;DISABLE ALL INTERRUPTS
004616 012767 000400 011100 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
004624 012767 004736 011066 MOV #2#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN-1
004632 000025 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
004640 012777 004000 011016 MOV #377,TDATA ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
004646 012777 000004 011002 MOV #4,@DHSCR ;SELECT LINE 4
004654 012777 177777 011004 MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
004662 012777 015756 010774 MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
004670 012777 033500 010764 MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
004676 052777 000003 010756 BIS #3,@DHLPR ;SET CHARACTER LENGTH FOR 10 BITS
004704 012777 000020 010756 MOV #20,@DHBR ;START TRANSMITTER
004712 105777 010740 1#: TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
004716 100375 BPL 1#
004720 017704 010734 MOV @DHNR,R4 ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
004724 012705 102377 MOV #102377,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
;AND LINE NUMBER AND CHARACTER
;STATUS IN HIGH BYTE
;ARE EXPECTED AND RECEIVED DATA THE SAME
004730 020504 CMP R5,R4
004732 001401 BEQ 2#
004734 HLT
004734 104000 EMT ;CHARACTER LENGTH, DATA
;OR LINE NUMBER ERROR
004736 104400 2#: SCOPE
000004 LENGTX=LENGTX+1
000011 LENGTC=LENGTC+1
000777 XCODE=XCODE+2+1
000005 LINEX=LINEX+1
000040 BITX=BITX+BITX
000000 LENGTX=0
000005 LENGTC=5
000037 XCODE=37
000004 .REPT 4
.NLIST
DATAR=LINEX+400+100000*XCODE
.LIST
CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
.NLIST
LENGTX=LENGTX+1
LENGTC=LENGTC+1
XCODE=XCODE+2+1
.LIST
.ENDR
DATAR=LINEX+400+100000*XCODE
004740 102437 CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 5

```

```

;CHARACTER LENGTH IS 5 BITS
;EXPECTED RECEIVED CHARACTER IS 37
;LINE SPEED IS 9600 BAUD

004740      000026      TS \XN,400,2#
004740 012767 00034^ 173030 T25:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
004746 012767 000400 010750      MOV  #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
004754 012767 005066 010736      MOV  #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV  #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

004762 012777 004000 010666      MOV  #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
004770 012767 000037 010760      MOV  #37,TDATA       ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
004776 012777 000005 010652      MOV  #5,@DHSCR       ;SELECT LINE 5
005004 012777 177777 010654      MOV  #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
005012 012777 015756 010644      MOV  #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
005020 012777 033500 010634      MOV  #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
005026 052777 000000 010626      BIS  #0,@DHLPR       ;SET CHARACTER LENGTH FOR 5 BITS
005034 012777 000040 010626      MOV  #40,@DHBAR      ;START TRANSMITTER
005042 105777 010610      1#:  TSTB  @DHSCR       ;WAIT TO RECEIVE CHARACTER
005046 100375      BPL  1#
005050 017704 010604      MOV  @DHNRC,R4       ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
005054 012705 102437      MOV  #102437,R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
005060 020504      CMP  R5,R4          ;STATUS IN HIGH BYTE
005062 001401      BEQ  2#             ;ARE EXPECTED AND RECEIVED DATA THE SAME
005064      HLT
005064 104000      EMT                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

005066 104400      2#:  SCOPE
000001      LENGTX=LENGTX+1
000006      LENGTC=LENGTC+1
000077      XCODE=XCODE+2+1
102477      DATAR=LINEX+400+100000+XCODE
005070      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 5
;CHARACTER LENGTH IS 6 BITS
;EXPECTED RECEIVED CHARACTER IS 77
;LINE SPEED IS 9600 BAUD

005070      TS \XN,400,2#
005070 012767 000340 172700 T26:  MOV  #340,PS           ;DISABLE ALL INTERRUPTS
005076 012767 000400 010620      MOV  #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
005104 012767 005216 010606      MOV  #2#,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV  #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

005112 012777 004000 010536      MOV  #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
005120 012767 000077 010630      MOV  #77,TDATA       ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
005126 012777 000005 010522      MOV  #5,@DHSCR       ;SELECT LINE 5

```

```

005134 012777 177777 010524      MOV      #-1, @DHBC      ;SET UP TO TRANSMIT 1 BYTE
005142 012777 015756 010514      MOV      @TDATA, @DHBA   ;SFT UP ADDRESS OF CHARACTER TO BE TRANSMITTED
005150 012777 033500 010504      MOV      @33500, @DHLPR  ;SET LINE SPEED FOR 9600 BAUD
005156 052777 000001 010476      BIS      @1, @DHLPR      ;SET CHARACTER LENGTH FOR 6 BITS
005164 012777 000040 010476      MOV      @40, @DHBAR     ;START TRANSMITTER
005172 105777 010460      1$: TSTB   @DHSCR         ;WAIT TO RECEIVE CHARACTER
005176 100375      BPL      1$
005200 017704 010454      MOV      @DHNRC, R4     ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
005204 012705 102477      MOV      @102477, R5    ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
005210 020504      CMP      R5, R4        ;STATUS IN HIGH BYTE
005212 001401      BEQ      2$           ;ARE EXPECTED AND RECEIVED DATA THE SAME
005214      HLT
005214 104000      EMT                   ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
005216 104400      2$: SCOPE
000002      LENGTX=LENGTX+1
000007      LENGTC=LENGTC+1
000177      XCODE=XCODE+2+1
102577      DATAR=LINEX+400+100000+XCODE
005220      CHART1 \LINEX, \BITX, \LENGTX, \LENGTC, \XCODE, \DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 5
                                ;CHARACTER LENGTH IS 7 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 177
                                ;LINE SPEED IS 9600 BAUD
005220      TS \XN, 400, 2$
005220 012767 000340 172550      T27: MOV      @340, PS    ;DISABLE ALL INTERRUPTS
005226 012767 000400 010470      MOV      @400, ICOUNT   ;SET UP FOR 400 ITERATIONS
005234 012767 005346 010456      MOV      @2$, ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      @, FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
005242 012777 004000 010406      MOV      @BIT11, @DHSCR  ;MASTER CLEAR INTERFACE
005250 012767 000177 010500      MOV      @177, TDATA     ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
005256 012777 000005 010372      MOV      @5, @DHSCR     ;SELECT LINE 5
005264 012777 177777 010374      MOV      #-1, @DHBC     ;SET UP TO TRANSMIT 1 BYTE
005272 012777 015756 010364      MOV      @TDATA, @DHBA  ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
005300 012777 033500 010354      MOV      @33500, @DHLPR ;SET LINE SPEED FOR 9600 BAUD
005306 052777 000002 010346      BIS      @2, @DHLPR     ;SET CHARACTER LENGTH FOR 7 BITS
005314 012777 000040 010346      MOV      @40, @DHBAR    ;START TRANSMITTER
005322 105777 010330      1$: TSTB   @DHSCR         ;WAIT TO RECEIVE CHARACTER
005326 100375      BPL      1$
005330 017704 010324      MOV      @DHNRC, R4     ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
005334 012705 102577      MOV      @102577, R5    ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
005340 020504      CMP      R5, R4        ;STATUS IN HIGH BYTE
005342 001401      BEQ      2$           ;ARE EXPECTED AND RECEIVED DATA THE SAME

```

```

005344          HLT          ;CHARACTER LENGTH, DATA
005344 104000    EMT          ;OR LINE NUMBER ERROR

005346 104400    2$: SCOPE
000003          LENGTX=LENGTX+1
000010          LENGTC=LENGTC+1
000377          XCODE=XCODE+2+1
102777          DATAR=LINEX*400+100000+XCODE
005350          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 5
;CHARACTER LENGTH IS 10 BITS
;EXPECTED RECEIVED CHARACTER IS 377
;LINE SPEED IS 9600 BAUD

005350          TS \XN,400,2$
005350 012767 000340 172420 T30: MOV #340,PS          ;DISABLE ALL INTERRUPTS
005356 012767 000400 010340 MOV #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
005364 012767 005476 010326 MOV #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

;IF NB <>
MOV #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3

.ENDC
XN=XN+1

005372 012777 004000 010256 MOV #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
005400 012767 000377 010350 MOV #377,TDATA      ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
005406 012777 000005 010242 MOV #5,@DHSCR       ;SELECT LINE 5
005414 012777 177777 010244 MOV #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
005422 012777 015756 010234 MOV #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
005430 012777 033500 010224 MOV #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
005436 052777 000003 010216 BIS #3,@DHLPR        ;SET CHARACTER LENGTH FOR 10 BITS
005444 012777 000040 010216 MOV #40,@DHBAR       ;START TRANSMITTER
005452 105777 010200 1$: TSTB @DHSCR          ;WAIT TO RECEIVE CHARACTER
005456 100375          BPL 1$
005460 017704 010174          MOV @DHNRC,R4          ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
; (R5)=EXPECTED CHARACTER IN LOW BYTE
;AND LINE NUMBER AND CHARACTER
;STATUS IN HIGH BYTE
;ARE EXPECTED AND RECEIVED DATA THE SAME

005464 012705 102777          MOV #102777,R5

005470 020504          CMP R5,R4
005472 001401          BEQ 2$
005474          HLT
005474 104000          EMT          ;CHARACTER LENGTH, DATA
;OR LINE NUMBER ERROR

005476 104400    2$: SCOPE
000004          LENGTX=LENGTX+1
000011          LENGTC=LENGTC+1
000777          XCODE=XCODE+2+1
000006          LINEX=LINEX+1
000100          BITX=BITX+BITX
000000          LENGTX=0
000005          LENGTC=5
000037          XCODE=37
000004          .REPT 4
;NLIST
DATAR=LINEX*400+100000+XCODE

```

```

.LIST
CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
.NLIST
LENGTX=LENGTX+1
LENGTC=LENGTC+1
XCODE=XCODE+2+1
.LIST
.ENDR
005500 103037 DATAR=LINEX*400+100000+XCODE
CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 6
;CHARACTER LENGTH IS 5 BITS
;EXPECTED RECEIVED CHARACTER IS 37
;LINE SPEED IS 9600 BAUD

005500 TS \XN,400,2$
005500 012767 000340 172270 T31: MOV #340,PS ;DISABLE ALL INTERRUPTS
005506 012767 000400 010210 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
005514 012767 005626 010176 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
005522 000032 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
005530 012777 004000 010126 MOV #37,TDATA ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
005536 012777 000006 010112 MOV #6,@DHSCR ;SELECT LINE 6
005544 012777 177777 010114 MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
005552 012777 015756 010104 MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
005560 012777 033500 010074 MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
005566 052777 000000 010066 BIS #0,@DHLPR ;SET CHARACTER LENGTH FOR 5 BITS
005574 012777 000100 010066 MOV #100,@DHBAR ;START TRANSMITTER
005602 105777 010050 1$: TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
005606 100375 BPL 1$
005610 017704 010044 MOV @DHNRC,R4 ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
005614 012705 103037 MOV #103037,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
;AND LINE NUMBER AND CHARACTER
;STATUS IN HIGH BYTE
005620 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME
005622 001401 BEQ 2$
005624 HLT ;CHARACTER LENGTH, DATA
005624 104000 EMT ;OR LINE NUMBER ERROR

005626 104400 2$: SCOPE
000001 LENGTX=LENGTX+1
000006 LENGTC=LENGTC+1
000077 XCODE=XCODE+2+1
103077 DATAR=LINEX*400+100000+XCODE
005630 - CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 6
;CHARACTER LENGTH IS 6 BITS
;EXPECTED RECEIVED CHARACTER IS 77

```

;LINE SPEED IS 9600 BAUD

```

005630      000033      TS \XN,400,2$
005630 012767 000340 172140 T32:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
005636 012767 000400 010060      MOV    #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
005644 012767 005756 010046      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

                .IF NB <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1

005652 012777 004000 007776      MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
005660 012767 000077 010070      MOV    #77,TDATA       ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
005666 012777 000006 007762      MOV    #6,@DHSCR       ;SELECT LINE 6
005674 012777 177777 007764      MOV    #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
005702 012777 015756 007754      MOV    #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
005710 012777 033500 007744      MOV    #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
005716 052777 000001 007736      BIS    #1,@DHLPR       ;SET CHARACTER LENGTH FOR 6 BITS
005724 012777 000100 007736      MOV    #100,@DHBAR     ;START TRANSMITTER
005732 105777 007720      1$:  TSTB   @DHSCR        ;WAIT TO RECEIVE CHARACTER
005736 100375      BPL    1$
005740 017704 007714      MOV    @DHNRC,R4       ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
005744 012705 103077      MOV    #103077,R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
005750 020504      CMP    R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME
005752 001401      BEQ    2$
005754      HLT
005754 104000      EMT                    ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

005756 104400      2$:  SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                DATAR=LINEX*400+100000+XCODE
005760      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 6
                                ;CHARACTER LENGTH IS 7 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 177
                                ;LINE SPEED IS 9600 BAUD

005760      TS \XN,400,2$
005760 012767 000340 172010 T33:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
005766 012767 000400 007730      MOV    #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
005774 012767 006106 007716      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

                .IF NB <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1

006002 012777 004000 007646      MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
006010 012767 000177 007740      MOV    #177,TDATA       ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
006016 012777 000006 007632      MOV    #6,@DHSCR       ;SELECT LINE 6
006024 012777 177777 007634      MOV    #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
006032 012777 015756 007624      MOV    #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED

```

```

006040 012777 033500 007614      MOV      #33500,@DHLPR      ;SET LINE SPEED FOR 9600 BAUD
006046 052777 000002 007606      BIS      #2,@DHLPR        ;SET CHARACTER LENGTH FOR 7 BITS
006054 012777 000100 007606      MOV      #100,@DHBAR      ;START TRANSMITTER
006062 105777 007570          1$:    TSTB      @DHSCR        ;WAIT TO RECEIVE CHARACTER
006066 100375          BPL      1$
006070 017704 007564          MOV      @DHNRC,R4        ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
006074 012705 103177          MOV      #103177,R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
006100 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME
006102 001401          BEQ      2$
006104          HLT
006104 104000          EMT        ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

006106 104400          2$:    SCOPE
000003          LENGTX=LENGTX+1
000010          LENGTC=LENGTC+1
000377          XCODE=XCODE+2+1
103377          DATAR=LINEX*400+100000+XCODE
006110          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 6
                                ;CHARACTER LENGTH IS 10 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 377
                                ;LINE SPEED IS 9600 BAUD

006110          TS \XN,400,2$
006110 012767 000340 171660      T34:    MOV      #340,PS      ;DISABLE ALL INTERRUPTS
006116 012767 000400 007600      MOV      #400,ICOUNT     ;SET UP FOR 400 ITERATIONS
006124 012767 006236 007566      MOV      #2$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1        ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
006132 000035          MOV      #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
006140 012767 000377 007516      MOV      #377,TDATA     ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
006146 012777 000006 007502      MOV      #6,@DHSCR      ;SELECT LINE 6
006154 012777 177777 007504      MOV      #-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
006162 012777 015756 007474      MOV      #TDATA,@DHBA   ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
006170 012777 033500 007464      MOV      #33500,@DHLPR  ;SET LINE SPEED FOR 9600 BAUD
006176 052777 000003 007456      BIS      #3,@DHLPR      ;SET CHARACTER LENGTH FOR 10 BITS
006204 012777 000100 007456      MOV      #100,@DHBAR    ;START TRANSMITTER
006212 105777 007440          1$:    TSTB      @DHSCR        ;WAIT TO RECEIVE CHARACTER
006216 100375          BPL      1$
006220 017704 007434          MOV      @DHNRC,R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
006224 012705 103377          MOV      #103377,R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
006230 020504          CMP      R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME
006232 001401          BEQ      2$
006234          HLT
006234 104000          EMT        ;CHARACTER LENGTH, DATA

```

```

006236 104400          2$: SCOPE                                ;OR LINE NUMBER ERROR
000004          LENGTX=LENGTX+1
000011          LENGTC=LENGTC+1
000777          XCODE=XCODE*2+1
000007          LINEX=LINEX+1
000200          BITX=BITX+BITX
000000          LENGTX=0
000005          LENGTC=5
000037          XCODE=37
000004          .REPT 4
                .NLIST
                DATAR=LINEX*400+100000+XCODE
                .LIST
                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                .NLIST
                LENGTX=LENGTX+1
                LENGTC=LENGTC+1
                XCODE=XCODE*2+1
                .LIST
                .ENDR
006240 103437          DATAR=LINEX*400+100000+XCODE
                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                ;CHARACTER LENGTH TEST
                ;TRANSMIT 1 CHARACTER ON LINE 7
                ;CHARACTER LENGTH IS 5 BITS
                ;EXPECTED RECEIVED CHARACTER IS 37
                ;LINE SPEED IS 9600 BAUD

006240 000036          TS \XN,400,2$
006240 012767 000340 171530 T35: MOV #340,PS                                ;DISABLE ALL INTERRUPTS
006246 012767 000400 007450      MOV #400,ICOUNT                          ;SET UP FOR 400 ITERATIONS
006254 012767 006366 007436      MOV #2$,ESCAPE                            ;SET UP TO ESCAPE TO NEXT TEST

                .IF NB <>
                MOV #,FREEZ1                                ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1

006262 012777 004000 007366      MOV #BIT11,@DHSCR                                ;MASTER CLEAR INTERFACE
006270 012767 000037 007460      MOV #37,TDATA                                  ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
006276 012777 000007 007352      MOV #7,@DHSCR                                    ;SELECT LINE 7
006304 012777 177777 007354      MOV #-1,@DHBC                                    ;SET UP TO TRANSMIT 1 BYTE
006312 012777 015756 007344      MOV #TDATA,@DHBA                                  ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
006320 012777 033500 007334      MOV #33500,@DHLPR                               ;SET LINE SPEED FOR 9600 BAUD
006326 052777 000000 007326      BIS #0,@DHLPR                                    ;SET CHARACTER LENGTH FOR 5 BITS
006334 012777 000200 007326      MOV #200,@DHBAR                                ;START TRANSMITTER
006342 105777 007310          1$: TSTB @DHSCR                                ;WAIT TO RECEIVE CHARACTER
006346 100375          BPL 1$
006350 017704 007304          MOV @DHNRC,R4                                ;(R4)=RECEIVED CHARACTER
                ;IN LOW BYTE, AND LINE NUMBER AND
                ;CHARACTER STATUS IN HIGH BYTE
006354 012705 103437          MOV #103437,R5                                ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                ;AND LINE NUMBER AND CHARACTER
                ;STATUS IN HIGH BYTE
006360 020504          CMP R5,R4                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
006362 001401          BEQ 2$
006364          HLT
                ;CHARACTER LENGTH, DATA

```

```

006364 104000          EMT
006366 104400          2$: SCOPE
000001          LENGTX=LENGTX+1
000006          LENGTC=LENGTC+1
000077          XCODE=XCODE+2+1
103477          DATAR=LINEX*400+100000+XCODE
006370          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 7
;CHARACTER LENGTH IS 6 BITS
;EXPECTED RECEIVED CHARACTER IS 77
;LINE SPEED IS 9600 BAUD

006370          TS \XN,400,2$
006370 012767 000340 171400 T36: MOV #340,PS ;DISABLE ALL INTERRUPTS
006376 012767 000400 007320 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
006404 012767 006516 007306 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
006412 012777 004000 007236 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
006420 012767 000077 007330 MOV #77,TDATA ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
006426 012777 000007 007222 MOV #7,@DHSCR ;SELECT LINE 7
006434 012777 177777 007224 MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
006442 012777 015756 007214 MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
006450 012777 033500 007204 MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
006456 052777 000001 007176 BIS #1,@DHLPR ;SET CHARACTER LENGTH FOR 6 BITS
006464 012777 000200 007176 MOV #200,@DHBAR ;START TRANSMITTER
006472 105777 007160 1$: TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
006476 100375 BPL 1$
006500 017704 007154 MOV @DHNRC,R4 ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
006504 012705 103477 MOV #103477,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
;AND LINE NUMBER AND CHARACTER
;STATUS IN HIGH BYTE
;ARE EXPECTED AND RECEIVED DATA THE SAME
006510 020504 CMP R5,R4
006512 001401 BEQ 2$
006514 HLT
006514 104000 EMT ;CHARACTER LENGTH, DATA
;OR LINE NUMBER ERROR

006516 104400          2$: SCOPE
000002          LENGTX=LENGTX+1
000007          LENGTC=LENGTC+1
000177          XCODE=XCODE+2+1
103577          DATAR=LINEX*400+100000+XCODE
006520          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 7
;CHARACTER LENGTH IS 7 BITS
;EXPECTED RECEIVED CHARACTER IS 177
;LINE SPEED IS 9600 BAUD

```

```

006520      000040      TS \XN,400,2$
006520 012767 000340 171250 T37:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
006526 012767 000400 007170      MOV    #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
006534 012767 006646 007156      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

006542 012777 004000 007106      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
006550 012767 000177 007200      MOV    #177,TDATA       ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
006556 012777 000007 007072      MOV    #7,@DHSCR        ;SELECT LINE 7
006564 012777 177777 007074      MOV    #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
006572 012777 015756 007064      MOV    #TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
006600 012777 033500 007054      MOV    #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
006606 052777 000002 007046      BIS    #2,@DHLPR        ;SET CHARACTER LENGTH FOR 7 BITS
006614 012777 000200 007046      MOV    #200,@DHBAR      ;START TRANSMITTER
006622 105777 007030      1$:  TSTB   @DHSCR        ;WAIT TO RECEIVE CHARACTER
006626 100375      BPL    1$
006630 017704 007024      MOV    @DHNRC,R4        ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
006634 012705 103577      MOV    #103577,R5       ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
006640 020504      CMP    R5,R4            ;STATUS IN HIGH BYTE
006642 001401      BEQ    2$              ;ARE EXPECTED AND RECEIVED DATA THE SAME
006644      HLT
006644 104000      EMT                    ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

006646 104400      2$:  SCOPE
000003      LENGTX=LENGTX+1
000010      LENGTC=LENGTC+1
000377      XCODE=XCODE+2+1
103777      DATAR=LINEX+400+100000+XCODE
006650      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 7
                                ;CHARACTER LENGTH IS 10 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 377
                                ;LINE SPEED IS 9600 BAUD

006650      TS \XN,400,2$
006650 012767 000340 171120 T40:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
006656 012767 000400 007040      MOV    #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
006664 012767 006776 007026      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

006672 012777 004000 006756      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
006700 012767 000377 007050      MOV    #377,TDATA       ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
006706 012777 000007 006742      MOV    #7,@DHSCR        ;SELECT LINE 7
006714 012777 177777 006744      MOV    #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
006722 012777 015756 006734      MOV    #TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
006730 012777 033500 006724      MOV    #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
006736 052777 000003 006716      BIS    #3,@DHLPR        ;SET CHARACTER LENGTH FOR 10 BITS

```

```

006744 012777 000200 006716      MOV    #200,@DHBAR      ;START TRANSMITTER
006752 105777 006700      1$:   TSTB   @DHSCR      ;WAIT TO RECEIVE CHARACTER
006756 100375      BPL    1$
006760 017704 006674      MOV    @DHNRC,R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
006764 012705 103777      MOV    #103777,R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
006770 020504      CMP    R5,R4          ;STATUS IN HIGH BYTE
006772 001401      BEQ    2$            ;ARE EXPECTED AND RECEIVED DATA THE SAME
006774      HLT
006774 104000      EMT
                                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR

006776 104400      2$:   SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                LINEX=LINEX+1
                                BITX=BITX+BITX
                                LENGTX=0
                                LENGTC=5
                                XCODE=37
                                .REPT 4
                                .NLIST
                                DATAR=LINEX*400+100000+XCODE
                                .LIST
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                .NLIST
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                .LIST
                                .ENDR
                                DATAR=LINEX*400+100000+XCODE
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

007000 104037      ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 10
                                ;CHARACTER LENGTH IS 5 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 37
                                ;LINE SPEED IS 9600 BAUD

007000      TS \XN,400,2$
007000 012767 000340 170770      T41:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
007006 012767 000400 006710      MOV    #400,ICOUNT     ;SET UP FOR 400 ITERATIONS
007014 012767 007126 006676      MOV    #2$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    #,FREEZ1
                                ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1

007022 012777 004000 006626      MOV    @BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
007030 012767 000037 006720      MOV    #37,TDATA      ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
007036 012777 000010 006612      MOV    #10,@DHSCR     ;SELECT LINE 10
007044 012777 177777 006614      MOV    #-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
007052 012777 015756 006604      MOV    $TDATA,@DHBA   ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
007060 012777 033500 006574      MOV    #33500,@DPR    ;SET LINE SPEED FOR 9600 BAUD

```



```

007256 104400          2#:  SCOPE
          000002      LENGTX=LENGTX+1
          000007      LENGTC=LENGTC+1
          000177      XCODE=XCODE+2+1
          104177      DATAR=LINEX+400+100000+XCODE
007260          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

          ;CHARACTER LENGTH TEST
          ;TRANSMIT 1 CHARACTER ON LINE 10
          ;CHARACTER LENGTH IS 7 BITS
          ;EXPECTED RECEIVED CHARACTER IS 177
          ;LINE SPEED IS 9600 BAUD

007260          TS \XN,400,2#
007260 012767 000340 170510 T43:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
007266 012767 000400 006430      MOV    #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
007274 012767 007406 006416      MOV    #2#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB  <>
          MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1
007302 000044          MOV    #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
007310 012777 004000 006346      MOV    #177,TDATA      ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
007310 012767 000177 006440      MOV    #10,@DHSCR      ;SELECT LINE 10
007316 012777 000010 006332      MOV    #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
007324 012777 177777 006334      MOV    #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
007332 012777 015756 006324      MOV    #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
007340 012777 033500 006314      BIS    #2,@DHLPR       ;SET CHARACTER LENGTH FOR 7 BITS
007346 052777 000002 006306      MOV    #400,@DHBAR    ;START TRANSMITTER
007354 012777 000400 006306      1#:  TSTB   @DHSCR       ;WAIT TO RECEIVE CHARACTER
007362 105777 006270          BPL    1#
007366 100375          MOV    @DHNR,R4
007370 017704 006264          ;(R4)=RECEIVED CHARACTER
          ;IN LOW BYTE, AND LINE NUMBER AND
          ;CHARACTER STATUS IN HIGH BYTE
007374 012705 104177          MOV    #104177,R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
          ;AND LINE NUMBER AND CHARACTER
          ;STATUS IN HIGH BYTE
          ;ARE EXPECTED AND RECEIVED DATA THE SAME
007400 020504          CMP    R5,R4
007402 001401          BEQ    2#
007404          HLT
007404 104000          EMT
          ;CHARACTER LENGTH, DATA
          ;OR LINE NUMBER ERROR

007406 104400          2#:  SCOPE
          000003      LENGTX=LENGTX+1
          000010      LENGTC=LENGTC+1
          000377      XCODE=XCODE+2+1
          104377      DATAR=LINEX+400+100000+XCODE
007410          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

          ;CHARACTER LENGTH TEST
          ;TRANSMIT 1 CHARACTER ON LINE 10
          ;CHARACTER LENGTH IS 10 BITS
          ;EXPECTED RECEIVED CHARACTER IS 377
          ;LINE SPEED IS 9600 BAUD

007410          TS \XN,400,2#
007410 012767 000340 170360 T44:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS

```

```

007416 012767 000400 006300      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
007424 012767 007536 006266      MOV      #2#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
007432 012777 004000 006216      MOV      @BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
007440 012767 000377 006310      MOV      #377,TDATA      ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
007446 012777 000010 006202      MOV      #10,@DHSCR      ;SELECT LINE 10
007454 012777 177777 006204      MOV      #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
007462 012777 015756 006174      MOV      @TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
007470 012777 033500 006164      MOV      #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
007476 052777 000003 006156      BIS      #3,@DHLPR       ;SET CHARACTER LENGTH FOR 10 BITS
007504 012777 000400 006156      MOV      #400,@DHBAR     ;START TRANSMITTER
007512 105777 006140 1#      TSTB    @DHSCR          ;WAIT TO RECEIVE CHARACTER
007516 100375 1#      BPL     1#
007520 017704 006134 1#      MOV     @DHNRC,R4

                                ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
007524 012705 104377      MOV     #104377,R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
007530 020504      CMP     R5,R4
007532 001401      BEQ     2#
007534      HLT
007534 104000      EMT

                                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
007536 104400      2#:    SCOPE
                                000004      LENGTX=LENGTX+1
                                000011      LENGTC=LENGTC+1
                                000777      XCODE=XCODE+2+1
                                000011      LINEX=LINEX+1
                                001000      BITX=BITX+BITX
                                000000      LENGTX=0
                                000005      LENGTC=5
                                000037      XCODE=37
                                000004      .REPT 4
                                .NLIST
                                DATAR=LINEX*400+100000*XCODE
                                .LIST
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                .NLIST
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                .LIST
                                .ENDR
                                DATAR=LINEX*400+100000*XCODE
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 11
                                ;CHARACTER LENGTH IS 5 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 37
                                ;LINE SPEED IS 9600 BAUD
007540      TS \XN,400,2#

```

```

007540 012767 000340 170230 T45:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
007546 012767 000400 006150      MOV  #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
007554 012767 007666 006136      MOV  #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
007562 012777 004000 006066      MOV  #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
007570 012767 000037 006160      MOV  #37,TDATA      ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
007576 012777 000011 006052      MOV  #11,@DHSCR     ;SELECT LINE 11
007604 012777 177777 006054      MOV  #-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
007612 012777 015756 006044      MOV  #TDATA,@DHBA   ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
007620 012777 033500 006034      MOV  #33500,@DHLPR  ;SET LINE SPEED FOR 9600 BAUD
007626 052777 000000 006026      BIS  #0,@DHLPR      ;SET CHARACTER LENGTH FOR 5 BITS
007634 012777 001000 006026      MOV  #1000,@DHBAR   ;START TRANSMITTER
007642 105777 006010              1$:  TSTB  @DHSCR        ;WAIT TO RECEIVE CHARACTER
007646 100375                      BPL  1$
007650 017704 006004              MOV  @DHNRC,R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
                                ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
007654 012705 104437              MOV  #104437,R5
007660 020504                      CMP  R5,R4
007662 001401                      BEQ  2$
007664                      HLT
007664 104000                      ENT
                                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
007666 104400              2$:  SCOPE
                                000001  LENGTX=LENGTX+1
                                000006  LENGTC=LENGTC+1
                                000077  XCODE=XCODE+2+1
                                104477  DATAR=LINEX+400+100000+XCODE
007670              CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 11
                                ;CHARACTER LENGTH IS 6 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 77
                                ;LINE SPEED IS 9600 BAUD
007670              TS \XN,400,2$
007670 012767 000340 170100 T46:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
007676 012767 000400 006020      MOV  #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
007704 012767 010016 006006      MOV  #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
007712 012777 004000 005736      MOV  #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
007720 012767 000077 006030      MOV  #77,TDATA      ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
007726 012777 000011 005722      MOV  #11,@DHSCR     ;SELECT LINE 11
007734 012777 177777 005724      MOV  #-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
007742 012777 015756 005714      MOV  #TDATA,@DHBA   ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
007750 012777 033500 005704      MOV  #33500,@DHLPR  ;SET LINE SPEED FOR 9600 BAUD
007756 052777 000001 005676      BIS  #1,@DHLPR      ;SET CHARACTER LENGTH FOR 6 BITS
007764 012777 001000 005676      MOV  #1000,@DHBAR   ;START TRANSMITTER

```

```

007772 105777 005660      1$:  TSTB  @DHSCR
007776 100375              BPL    1$
010000 017704 005654      MOV    @DHNRC,R4
                                ;WAIT TO RECEIVE CHARACTER
                                ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
010004 012705 104477      MOV    @104477,R5
                                ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
010010 020504              CMP    R5,R4
010012 001401              BEQ    2$
010014 104000              HLT
010014 104000              EMT
                                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
010016 104400      2$:  SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                DATAR=LINEX+400+100000+XCODE
010020      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 11
                                ;CHARACTER LENGTH IS 7 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 177
                                ;LINE SPEED IS 9600 BAUD
010020      TS \XN,400,2$
010020 012767 000340 167750 T47:  MOV    @340,PS
010026 012767 000400 005670      MOV    @400,ICOUNT
010034 012767 010146 005656      MOV    @2$,ESCAPE
                                ;DISABLE ALL INTERRUPTS
                                ;SET UP FOR 400 ITERATIONS
                                ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV    @,FREEZ1
                                ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
010042 012777 004000 005606      MOV    @BIT11,@DHSCR
010050 012767 000177 005700      MOV    @177,TDATA
010056 012777 000011 005572      MOV    @11,@DHSCR
010064 012777 177777 005574      MOV    @-1,@DHBC
010072 012777 015756 005564      MOV    @TDATA,@DHBA
010100 012777 033500 005554      MOV    @33500,@DHLPR
010106 052777 000002 005546      BIS    @2,@DHLPR
010114 012777 001000 005546      MOV    @1000,@DHBAR
010122 105777 005530      1$:  TSTB  @DHSCR
010126 100375              BPL    1$
010130 017704 005524      MOV    @DHNRC,R4
                                ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
010134 012705 104577      MOV    @104577,R5
                                ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
010140 020504              CMP    R5,R4
010142 001401              BEQ    2$
010144 104000              HLT
010144 104000              EMT
                                ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
010146 104400      2$:  SCOPE
                                LENGTX=LENGTX+1

```

```

000010          LENGTC=LENGTC+1
000377          XCODE=XCODE+2+1
104777          DATAR=LINEX*400+100000+XCODE
010150          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 11
;CHARACTER LENGTH IS 10 BITS
;EXPECTED RECEIVED CHARACTER IS 377
;LINE SPEED IS 9600 BAUD

010150          000051          TS \XN,400,2$
010150 012767 000340 167620 T50:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
010156 012767 000400 005540      MOV    #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
010164 012767 010276 005526      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST

          .IF NB    <>
          MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1

010172 012777 004000 005456      MOV    #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
010200 012767 000377 005550      MOV    #377,TDATA        ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
010206 012777 000011 005442      MOV    #11,@DHSCR        ;SELECT LINE 11
010214 012777 177777 005444      MOV    #-1,@DHBC         ;SET UP TO TRANSMIT 1 BYTE
010222 012777 015756 005434      MOV    #TDATA,@DHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
010230 012777 033500 005424      MOV    #33500,@DHLPR     ;SET LINE SPEED FOR 9600 BAUD
010236 052777 000003 005416      BIS    #3,@DHLPR         ;SET CHARACTER LENGTH FOR 10 BITS
010244 012777 001000 005416      MOV    #1000,@DHBAR     ;START TRANSMITTER
010252 105777 005400          1$:  TSTB   @DHSCR          ;WAIT TO RECEIVE CHARACTER
010256 100375          BPL    1$
010260 017704 005374          MOV    @DHNRC,R4

; (R4)=RECEIVED CHARACTER
; IN LOW BYTE, AND LINE NUMBER AND
; CHARACTER STATUS IN HIGH BYTE
; (R5)=EXPECTED CHARACTER IN LOW BYTE
; AND LINE NUMBER AND CHARACTER
; STATUS IN HIGH BYTE
; ARE EXPECTED AND RECEIVED DATA THE SAME

010264 012705 104777          MOV    #104777,R5

010270 020504          CMP    R5,R4
010272 001401          BEQ   2$
010274          HLT
010274 104000          EMT

; CHARACTER LENGTH, DATA
; OR LINE NUMBER ERROR

010276 104400          2$:  SCOPE
          000004          LENGTX=LENGTX+1
          000011          LENGTC=LENGTC+1
          000777          XCODE=XCODE+2+1
          000012          LINEX=LINEX+1
          002000          BITX=BITX+BITX
          000000          LENGTX=0
          000005          LENGTC=5
          000037          XCODE=37
          000004          .REPT 4
          .NLIST
          DATAR=LINEX*400+100000+XCODE
          .LIST
          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
          .NLIST
          LENGTX=LENGTX+1
          LENGTC=LENGTC+1

```

```

                                XCODE=XCODE*2+1
                                .LIST
                                .ENDR
010300 105037 DATAR=LINEX*400+100000+XCODE
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 12
                                ;CHARACTER LENGTH IS 5 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 37
                                ;LINE SPEED IS 9600 BAUD

010300 TS \XN,400,2#
010300 012767 000340 167470 T51: MOV #340,PS ;DISABLE ALL INTERRUPTS
010306 012767 000400 005410 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
010314 012767 010426 005376 MOV #2#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB <>
                                MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1

010322 000052 012777 004000 005326 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
010330 012767 000037 005420 MOV #37,TDATA ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
010336 012777 000012 005312 MOV #12,@DHSCR ;SELECT LINE 12
010344 012777 177777 005314 MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
010352 012777 015756 005304 MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
010360 012777 033500 005274 MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
010366 052777 000000 005266 BIS #0,@DHLPR ;SET CHARACTER LENGTH FOR 5 BITS
010374 012777 002000 005266 MOV #2000,@DHBAR ;START TRANSMITTER
010402 105777 005250 1#: TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
010406 100375 BPL 1#
010410 017704 005244 MOV @DHNRC,R4 ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
010414 012705 105037 MOV #105037,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
010420 020504 CMP R5,R4 ;STATUS IN HIGH BYTE
010422 001401 BEQ 2# ;ARE EXPECTED AND RECEIVED DATA THE SAME
010424 HLT ;CHARACTER LENGTH, DATA
010424 104000 EMT ;OR LINE NUMBER ERROR

010426 104400 2#: SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE*2+1
                                DATAR=LINEX*400+100000+XCODE
010430 105077 CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 12
                                ;CHARACTER LENGTH IS 6 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 77
                                ;LINE SPEED IS 9600 BAUD

010430 TS \XN,400,2#
010430 012767 000340 167340 T52: MOV #340,PS ;DISABLE ALL INTERRUPTS
010436 012767 000400 005260 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS

```

```

010444 012767 010556 005246      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
010452 012777 004000 005176      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
010460 012767 000077 005270      MOV      #77,TDATA      ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
010466 012777 000012 005162      MOV      #12,@DHSCR      ;SELECT LINE 12
010474 012777 177777 005164      MOV      #-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
010502 012777 015756 005154      MOV      #TDATA,@DHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
010510 012777 033500 005144      MOV      #33500,@DHLPR      ;SET LINE SPEED FOR 9600 BAUD
010516 052777 000001 005136      BIS      #1,@DHLPR      ;SET CHARACTER LENGTH FOR 6 BITS
010524 012777 002000 005136      MOV      #2000,@DHBAR      ;START TRANSMITTER
010532 105777 005120      1$: TSTB      @DHSCR      ;WAIT TO RECEIVE CHARACTER
010536 100375      BPL      1$
010540 017704 005114      MOV      @DHNRC,R4      ;(R4)=RECEIVED CHARACTER
      ;IN LOW BYTE, AND LINE NUMBER AND
      ;CHARACTER STATUS IN HIGH BYTE
010544 012705 105077      MOV      #105077,R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
      ;AND LINE NUMBER AND CHARACTER
      ;STATUS IN HIGH BYTE
010550 020504      CMP      R5,R4      ;ARE EXPECTED AND RECEIVED DATA THE SAME
010552 001401      BEQ      2$
010554      HLT
010554 104000      EMT      ;CHARACTER LENGTH, DATA
      ;OR LINE NUMBER ERROR
010556 104400      2$: SCOPE
      000002      LENGTX=LENGTX+1
      000007      LENGTC=LENGTC+1
      000177      XCODE=XCODE+2+1
      105177      DATAR=LINEX+400+100000+XCODE
010560      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
      ;CHARACTER LENGTH TEST
      ;TRANSMIT 1 CHARACTER ON LINE 12
      ;CHARACTER LENGTH IS 7 BITS
      ;EXPECTED RECEIVED CHARACTER IS 177
      ;LINE SPEED IS 9600 BAUD
010560      TS \XN,400,2$
010560 012767 000340 167210      T53: MOV      #340,PS      ;DISABLE ALL INTERRUPTS
010566 012767 000400 005130      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
010574 012767 010706 005116      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <>
      MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
010602 012777 004000 005046      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
010610 012767 000177 005140      MOV      #177,TDATA      ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
010616 012777 000012 005032      MOV      #12,@DHSCR      ;SELECT LINE 12
010624 012777 177777 005034      MOV      #-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
010632 012777 015756 005024      MOV      #TDATA,@DHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
010640 012777 033500 005014      MOV      #33500,@DHLPR      ;SET LINE SPEED FOR 9600 BAUD
010646 052777 000002 005006      BIS      #2,@DHLPR      ;SET CHARACTER LENGTH FOR 7 BITS
010654 012777 002000 005006      MOV      #2000,@DHBAR      ;START TRANSMITTER
010662 105777 004770      1$: TSTB      @DHSCR      ;WAIT TO RECEIVE CHARACTER
010666 100375      BPL      1$

```

```

010670 017704 004764      MOV      @DHNR,R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
010674 012705 105177      MOV      #105177,R5    ;CHARACTER STATUS IN HIGH BYTE
                                ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
010700 020504      CMP      R5,R4
010702 001401      BEQ      2#
010704      HLT
010704 104000      EMT      ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
010706 104400      2#:      SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                DATAR=LINEX+400+100000+XCODE
010710 105377      CHART1  \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 12
                                ;CHARACTER LENGTH IS 10 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 377
                                ;LINE SPEED IS 9600 BAUD

010710      TS \XN,400,2#
010710 012767 000340 167060 T54:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
010716 012767 000400 005000      MOV      #400,ICOUNT    ;SET UP FOR 400 ITERATIONS
010724 012767 011036 004766      MOV      #2#,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
010732 012777 004000 004716      MOV      @BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
010740 012767 000377 005010      MOV      #377,TDATA    ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
010746 012777 000012 004702      MOV      #12,@DHSCR    ;SELECT LINE 12
010754 012777 177777 004704      MOV      #-1,@DHBC     ;SET UP TO TRANSMIT 1 BYTE
010762 012777 015756 004674      MOV      @TDATA,@DHBA  ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
010770 012777 033500 004664      MOV      #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
010776 052777 000003 004656      BIS      #3,@DHLPR     ;SET CHARACTER LENGTH FOR 10 BITS
011004 012777 002000 004656      MOV      #2000,@DHBAR  ;START TRANSMITTER
011012 105777 004640      1#:  TSTB     @DHSCR      ;WAIT TO RECEIVE CHARACTER
011016 100375      BPL      1#
011020 017704 004634      MOV      @DHNR,R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
011024 012705 105377      MOV      #105377,R5    ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
011030 020504      CMP      R5,R4
011032 001401      BEQ      2#
011034      HLT
011034 104000      EMT      ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
011036 104400      2#:      SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1

```

```

000013          LINEX=LINEX+1
004000          BITX=BITX+BITX
000000          LENGTX=0
000005          LENGTC=5
000037          XCODE=37
000004          .REPT 4
                .NLIST
                DATAR=LINEX*400+100000+XCODE
                .LIST
                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                .NLIST
                LENGTX=LENGTX+1
                LENGTC=LENGTC+1
                XCODE=XCODE+2+1
                .LIST
                .ENDR
011040 105437   DATAR=LINEX*400+100000+XCODE
                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                ;CHARACTER LENGTH TEST
                ;TRANSMIT 1 CHARACTER ON LINE 13
                ;CHARACTER LENGTH IS 5 BITS
                ;EXPECTED RECEIVED CHARACTER IS 37
                ;LINE SPEED IS 9600 BAUD

011040          TS \XN,400,2$
011040 012767 000340 166730 T55:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
011046 012767 000400 004650      MOV    #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
011054 012767 011166 004636      MOV    #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                .IF NB <>
                MOV    #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
                .ENDC
                XN=XN+1
011062 012777 004000 004566      MOV    #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
011070 012767 000037 004660      MOV    #37,TDATA      ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
011076 012777 000013 004552      MOV    #13,@DHSCR     ;SELECT LINE 13
011104 012777 177777 004554      MOV    #-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
011112 012777 015756 004544      MOV    #TDATA,@DHBA   ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
011120 012777 033500 004534      MOV    #33500,@DHLPR  ;SET LINE SPEED FOR 9600 BAUD
011126 052777 000000 004526      BIS    #0,@DHLPR      ;SET CHARACTER LENGTH FOR 5 BITS
011134 012777 004000 004526      MOV    #4000,@DHBAR   ;START TRANSMITTER
011142 105777 004510 1$:      TSTB  @DHSCR          ;WAIT TO RECEIVE CHARACTER
011146 100375      BPL    1$
011150 017704 004504      MOV    @DHNR,R4
                ;(R4)=RECEIVED CHARACTER
                ;IN LOW BYTE, AND LINE NUMBER AND
                ;CHARACTER STATUS IN HIGH BYTE
011154 012705 105437      MOV    #105437,R5    ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                ;AND LINE NUMBER AND CHARACTER
                ;STATUS IN HIGH BYTE
                ;ARE EXPECTED AND RECEIVED DATA THE SAME
011160 020504      CMP    R5,R4
011162 001401      BEQ   2$
011164          HLT
011164 104000      EMT
                ;CHARACTER LENGTH, DATA
                ;OR LINE NUMBER ERROR
011166 104400      2$:  SCOPE
                000001  LENGTX=LENGTX+1
                000006  LENGTC=LENGTC+1

```

```

000077
105477
011170
XCODE=XCODE*2+1
DATAR=LINEX*400+100000+XCODE
CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 13
;CHARACTER LENGTH IS 6 BITS
;EXPECTED RECEIVED CHARACTER IS 77
;LINE SPEED IS 9600 BAUD

011170 012767 000340 166600 TS \XN,400,2$
011170 012767 000400 004520 T56: MOV #340,PS ;DISABLE ALL INTERRUPTS
011204 012767 011316 004506 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1

011212 012777 004000 004436 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
011220 012767 000077 004530 MOV #77,TDATA ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
011226 012777 000013 004422 MOV #13,@DHSCR ;SELECT LINE 13
011234 012777 177777 004424 MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
011242 012777 015756 004414 MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
011250 012777 033500 004404 MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
011256 052777 000001 004376 BIS #1,@DHLPR ;SET CHARACTER LENGTH FOR 6 BITS
011264 012777 004000 004376 MOV #4000,@DHBAR ;START TRANSMITTER
011272 105777 004360 1$: TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
011276 100375 BPL 1$
011300 017704 004354 MOV @DHNRC,R4 ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
;CHARACTER STATUS IN HIGH BYTE
;AND LINE NUMBER AND CHARACTER
;STATUS IN HIGH BYTE
;ARE EXPECTED AND RECEIVED DATA THE SAME

011304 012705 105477 MOV #105477,R5

011310 020504 CMP R5,R4
011312 001401 BEQ 2$
011314 HLT
011314 104000 EMT ;CHARACTER LENGTH, DATA
;OR LINE NUMBER ERROR

011316 104400 2$: SCOPE
000002 LENGTX=LENGTX+1
000007 LENGTC=LENGTC+1
000177 XCODE=XCODE*2+1
105577 DATAR=LINEX*400+100000+XCODE
011320 CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 13
;CHARACTER LENGTH IS 7 BITS
;EXPECTED RECEIVED CHARACTER IS 177
;LINE SPEED IS 9600 BAUD

011320 012767 000340 166450 TS \XN,400,2$
011320 012767 000400 004370 T57: MOV #340,PS ;DISABLE ALL INTERRUPTS
011326 012767 000400 004370 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
011334 012767 011446 004356 MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>

```



```

011564 012705 105777          MOV      #105777,R5          ;CHARACTER STATUS IN HIGH BYTE
                                          ;(R5)-EXPECTED CHARACTER IN LOW BYTE
                                          ;AND LINE NUMBER AND CHARACTER
011570 020504          CMP      R5,R4          ;STATUS IN HIGH BYTE
011572 001401          BEQ      2#           ;ARE EXPECTED AND RECEIVED DATA THE SAME
011574          HLT
011574 104000          EMT          ;CHARACTER LENGTH, DATA
                                          ;OR LINE NUMBER ERROR

011576 104400          2#:      SCOPE
000004          LENGTX=LENGTX+1
000011          LENGTC=LENGTC+1
000777          XCODE=XCODE+2+1
000014          LINEX=LINEX+1
010000          BITX=BITX+BITX
000000          LENGTX=0
000005          LENGTC=5
000037          XCODE=37
000004          .REPT 4
          .NLIST
          DATAR=LINEX*400+100000*XCODE
          .LIST
          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
          .NLIST
          LENGTX=LENGTX+1
          LENGTC=LENGTC+1
          XCODE=XCODE+2+1
          .LIST
          .ENDR
          DATAR=LINEX*400+100000*XCODE
011600 106037          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

          ;CHARACTER LENGTH TEST
          ;TRANSMIT 1 CHARACTER ON LINE 14
          ;CHARACTER LENGTH IS 5 BITS
          ;EXPECTED RECEIVED CHARACTER IS 37
          ;LINE SPEED IS 9600 BAUD

011600          TS \XN,400,2#
011600 012767 000340 166170 T61:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
011606 012767 000400 004110      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
011614 012767 011726 004076      MOV      #2#,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <>
          MOV      #,FREEZ1          ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1

011622 012777 004000 004026      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
011630 012767 000037 004120      MOV      #37,TDATA          ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
011636 012777 000014 004012      MOV      #14,@DHSCR          ;SELECT LINE 14
011644 012777 177777 004014      MOV      #-1,@DHBC          ;SET UP TO TRANSMIT 1 BYTE
011652 012777 015756 004004      MOV      #TDATA,@DHBA          ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
011660 012777 033500 003774      MOV      #33500,@DHLPR          ;SET LINE SPEED FOR 9600 BAUD
011666 052777 000000 003766      BIS      #0,@DHLPR          ;SET CHARACTER LENGTH FOR 5 BITS
011674 012777 010000 003766      MOV      #10000,@DHBAR          ;START TRANSMITTER
011702 105777 003750          1#:  TSTB      @DHSCR          ;WAIT TO RECEIVE CHARACTER
011706 100375          BPL      1#
011710 017704 003744          MOV      @DHNR,R4          ;(R4)=RECEIVED CHARACTER

```


012060

CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

```

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 14
;CHARACTER LENGTH IS 7 BITS
;EXPECTED RECEIVED CHARACTER IS 177
;LINE SPEED IS 9600 BAUD

```

```

012060 012060 012767 000340 165710 TS \XN,400,2#
012066 012767 000400 003630 T63: MOV #340,PS ;DISABLE ALL INTERRUPTS
012074 012767 012206 003616 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
MOV #2#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
012102 000064 012777 004000 003546 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
012110 012767 000177 003640 MOV #177,TDATA ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
012116 012777 000014 003532 MOV #14,@DHSCR ;SELECT LINE 14
012124 012777 177777 003534 MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
012132 012777 015756 003524 MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
012140 012777 033500 003514 MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
012146 052777 000002 003506 BIS #2,@DHLPR ;SET CHARACTER LENGTH FOR 7 BITS
012154 012777 010000 003506 MOV #10000,@DHBAR ;START TRANSMITTER
012162 105777 003470 1# TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
012166 100375 1# BPL 1#
012170 017704 003464 MOV @DHNR,R4 ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
012174 012705 106177 MOV #106177,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
;AND LINE NUMBER AND CHARACTER
;STATUS IN HIGH BYTE
012200 020504 CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME
012202 001401 BEQ 2#
012204 HLT ;CHARACTER LENGTH, DATA
012204 104000 EMT ;OR LINE NUMBER ERROR
012206 104400 2# SCOPE
000003 LENGTX=LENGTX+1
000010 LENGTC=LENGTC+1
000377 XCODE=XCODE+2+1
106377 DATAR=LINEX+400+100000+XCODE
012210 CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 14
;CHARACTER LENGTH IS 10 BITS
;EXPECTED RECEIVED CHARACTER IS 377
;LINE SPEED IS 9600 BAUD

```

```

012210 012210 012767 000340 165560 TS \XN,400,2#
012216 012767 000400 003500 T64: MOV #340,PS ;DISABLE ALL INTERRUPTS
012224 012767 012336 003466 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
MOV #2#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC

```

```

012232 000065      XN=XN-1
012232 012777 004000 003416      MOV      @BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
012240 012767 000377 003510      MOV      @377,TDATA      ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
012246 012777 000014 003402      MOV      @14,@DHSCR      ;SELECT LINE 14
012254 012777 177777 003404      MOV      @-1,@DHBC      ;SET UP TO TRANSMIT 1 BYTE
012262 012777 015756 003374      MOV      @TDATA,@DHBA      ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
012270 012777 033500 003364      MOV      @33500,@DHLPR      ;SET LINE SPEED FOR 9600 BAUD
012276 052777 000003 003356      BIS      @3,@DHLPR      ;SET CHARACTER LENGTH FOR 10 BITS
012304 012777 010000 003356      MOV      @10000,@DHBAR      ;START TRANSMITTER
012312 105777 003340      1$:      TSTB      @DHSCR      ;WAIT TO RECEIVE CHARACTER
012316 100375      BPL      1$
012320 017704 003334      MOV      @DHNRC,R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
012324 012705 106377      MOV      @106377,R5      ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
012330 020504      CMP      R5,R4      ;ARE EXPECTED AND RECEIVED DATA THE SAME
012332 001401      BEQ      2$
012334      HLT
012334 104000      EMT      ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
012336 104400      2$:      SCOPE
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                LINEX=LINEX+1
                                BITX=BITX+BITX
                                LENGTX=0
                                LENGTC=5
                                XCODE=37
                                .REPT 4
                                .NLIST
                                DATAR=LINEX+400+100000+XCODE
                                .LIST
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                .NLIST
                                LENGTX=LENGTX+1
                                LENGTC=LENGTC+1
                                XCODE=XCODE+2+1
                                .LIST
                                .ENDR
                                DATAR=LINEX+400+100000+XCODE
                                CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 15
                                ;CHARACTER LENGTH IS 5 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 37
                                ;LINE SPEED IS 9600 BAUD
012340      TS \XN,400,2$
012340 012767 000340 165430      T65:      MOV      @340,PS      ;DISABLE ALL INTERRUPTS
012346 012767 000400 003350      MOV      @400,ICOUNT      ;SET UP FOR 400 ITERATIONS
012354 012767 012466 003336      MOV      @2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      @,FREEZ1      ;SET UP TO LOOP WITH DATA

```



```

012604 012705 106477          MOV      #106477,R5          ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                           ;AND LINE NUMBER AND CHARACTER
012610 020504          CMP      R5,R4          ;STATUS IN HIGH BYTE
012612 001401          BEQ      2$            ;ARE EXPECTED AND RECEIVED DATA THE SAME
012614          HLT
012614 104000          EMT
                                           ;CHARACTER LENGTH, DATA
                                           ;OR LINE NUMBER ERROR

012616 104400          2$:      SCOPE
000002          LENGTX=LENGTX+1
000007          LENGTC=LENGTC+1
000177          XCODE=XCODE*2+1
106577          DATAR=LINEX*400+100000+XCODE
012620          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                                           ;CHARACTER LENGTH TEST
                                           ;TRANSMIT 1 CHARACTER ON LINE 15
                                           ;CHARACTER LENGTH IS 7 BITS
                                           ;EXPECTED RECEIVED CHARACTER IS 177
                                           ;LINE SPEED IS 9600 BAUD

012620          TS \XN,400,2$
012620 012767 000340 165150 T67:    MOV      #340,PS          ;DISABLE ALL INTERRUPTS
012626 012767 000400 003070      MOV      #400,ICOUNT      ;SET UP FOR 400 ITERATIONS
012634 012767 012746 003056      MOV      #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                           .IF NB <>
                                           MOV      #,FREEZ1        ;SET UP TO LOOP WITH DATA          ; 3
                                           .ENDC
                                           XN=XN+1

012642 012777 004000 003006      MOV      #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
012650 012767 000177 003100      MOV      #177,TDATA      ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)
012656 012777 000015 002772      MOV      #15,@DHSCR      ;SELECT LINE 15
012664 012777 177777 002774      MOV      #-1,@DHBC       ;SET UP TO TRANSMIT 1 BYTE
012672 012777 015756 002764      MOV      #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
012700 012777 033500 002754      MOV      #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
012706 052777 000002 002746      MOV      #2,@DHLPR       ;SET CHARACTER LENGTH FOR 7 BITS
012714 012777 020000 002746      MOV      #20000,@DHBAR   ;START TRANSMITTER
012722 105777 002730          1$:    TSTB    @DHSCR      ;WAIT TO RECEIVE CHARACTER
012726 100375          BPL      1$
012730 017704 002724          MOV      @DHNR,R4
                                           ;(R4)=RECEIVED CHARACTER
                                           ;IN LOW BYTE, AND LINE NUMBER AND
012734 012705 106577          MOV      #106577,R5      ;CHARACTER STATUS IN HIGH BYTE
                                           ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                           ;AND LINE NUMBER AND CHARACTER
012740 020504          CMP      R5,R4          ;STATUS IN HIGH BYTE
012742 001401          BEQ      2$            ;ARE EXPECTED AND RECEIVED DATA THE SAME
012744          HLT
012744 104000          EMT
                                           ;CHARACTER LENGTH, DATA
                                           ;OR LINE NUMBER ERROR

012746 104400          2$:      SCOPE
000003          LENGTX=LENGTX+1
000010          LENGTC=LENGTC+1
000377          XCODE=XCODE*2+1
106777          DATAR=LINEX*400+100000+XCODE
012750          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

```

```

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 15
;CHARACTER LENGTH IS 10 BITS
;EXPECTED RECEIVED CHARACTER IS 377
;LINE SPEED IS 9600 BAUD

012750
012750 012767 000340 165020 TS \XN,400,2$
012756 012767 000400 002740 T70: MOV #340,PS ;DISABLE ALL INTERRUPTS
012764 012767 013076 002726 MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
MOV #2$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <>
MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1

012772 000071
012772 012777 004000 002656 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
013000 012767 000377 002750 MOV #377,TDATA ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
013006 012777 000015 002642 MOV #15,@DHSCR ;SELECT LINE 15
013014 012777 177777 002644 MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
013022 012777 015756 002634 MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
013030 012777 033500 002624 MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
013036 052777 000003 002616 BIS #3,@DHLPR ;SET CHARACTER LENGTH FOR 10 BITS
013044 012777 020000 002616 MOV #20000,@DHBAR ;START TRANSMITTER
013052 105777 002600 1$: TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
013056 100375 BPL 1$
013060 017704 002574 MOV @DHNRC,R4 ;(R4)=RECEIVED CHARACTER
;IN LOW BYTE, AND LINE NUMBER AND
;CHARACTER STATUS IN HIGH BYTE
013064 012705 106777 MOV #106777,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
;AND LINE NUMBER AND CHARACTER
013070 020504 CMP R5,R4 ;STATUS IN HIGH BYTE
013072 001401 BEQ 2$ ;ARE EXPECTED AND RECEIVED DATA THE SAME
013074 HLT ;CHARACTER LENGTH, DATA
013074 104000 EMT ;OR LINE NUMBER ERROR

013076 104400 2$: SCOPE
000004 LENGTX=LENGTX+1
000011 LENGTC=LENGTC+1
000777 XCODE=XCODE+2+1
000016 LINEX=LINEX+1
040000 BITX=BITX+BITX
000000 LENGTX=0
000005 LENGTC=5
000037 XCODE=37
000004 .REPT 4
.NLIST
DATAR=LINEX*400+100000+XCODE
.LIST
CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
.NLIST
LENGTX=LENGTX+1
LENGTC=LENGTC+1
XCODE=XCODE+2+1
.LIST
.ENDR
DATAR=LINEX*400+100000+XCODE
013100 107037 CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

```

```

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 16
;CHARACTER LENGTH IS 5 BITS
;EXPECTED RECEIVED CHARACTER IS 37
;LINE SPEED IS 9600 BAUD

013100      013100 012767 000340 164670 TS \XN,400,2$
013100      012767 000340 164670 T71:  MOV   #340,PS           ;DISABLE ALL INTERRUPTS
013106      012767 000400 002610          MOV   #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
013114      012767 013226 002576          MOV   #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

013122      000072 012777 004000 002526          MOV   #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
013130      012767 000037 002620          MOV   #37,TDATA       ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
013136      012777 000016 002512          MOV   #16,@DHSCR      ;SELECT LINE 16
013144      012777 177777 002514          MOV   #-1,@DMBC       ;SET UP TO TRANSMIT 1 BYTE
013152      012777 015756 002504          MOV   #TDATA,@DHBA    ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
013160      012777 033500 002474          MOV   #33500,@DHLPR   ;SET LINE SPEED FOR 9600 BAUD
013166      052777 000000 002466          BIS   #0,@DHLPR       ;SET CHARACTER LENGTH FOR 5 BITS
013174      012777 040000 002466          MOV   #40000,@DHBAR   ;START TRANSMITTER
013202      105777 002450 1$:  TSTB  @DHSCR          ;WAIT TO RECEIVE CHARACTER
013206      100375          BPL   1$
013210      017704 002444          MOV   @DHNRC,R4       ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
013214      012705 107037          MOV   #107037,R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
013220      020504          CMP   R5,R4          ;ARE EXPECTED AND RECEIVED DATA THE SAME
013222      001401          BEQ   2$
013224          HLT
013224      104000          EMT
                                ;CHARACTER LENGTH, DATA
013226      104400          2$:  SCOPE
                                000001  LENGTX=LENGTX+1
                                000006  LENGTC=LENGTC+1
                                000077  XCODE=XCODE+2+1
                                107077  DATAR=LINEX+400+100000+XCODE
013230          CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 16
                                ;CHARACTER LENGTH IS 6 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 77
                                ;LINE SPEED IS 9600 BAUD

013230      013230 012767 000340 164540 TS \XN,400,2$
013230      012767 000340 164540 T72:  MOV   #340,PS           ;DISABLE ALL INTERRUPTS
013236      012767 000400 002460          MOV   #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
013244      012767 013356 002446          MOV   #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <>
                                MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
000073

```



```

013500 020504
013502 001401
013504
013504 104000
013506 104400
000003
000010
000377
107377
013510

                CMP      R5,R4
                BEQ      2$
                HLT
                EMT

                ;STATUS IN HIGH BYTE
                ;ARE EXPECTED AND RECEIVED DATA THE SAME

                ;CHARACTER LENGTH, DATA

                ;OR LINE NUMBER ERROR

2$:   SCOPE
      LENGTX=LENGTX+1
      LENGTC=LENGTC+1
      XCODE=XCODE+2+1
      DATAR=LINEX+400+100000+XCODE
      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

                ;CHARACTER LENGTH TEST
                ;TRANSMIT 1 CHARACTER ON LINE 16
                ;CHARACTER LENGTH IS 10 BITS
                ;EXPECTED RECEIVED CHARACTER IS 377
                ;LINE SPEED IS 9600 BAUD

013510
013510 012767 000340 164260
013516 012767 000400 002200
013524 012767 013636 002166

                TS \XN,400,2$
T74:  MOV      #340,PS
                MOV      #400,ICOUNT
                MOV      #2$,ESCAPE
                .IF NB <>
                MOV      #,FREEZ1
                .ENDC
                XN=XN+1

                ;DISABLE ALL INTERRUPTS
                ;SET UP FOR 400 ITERATIONS
                ;SET UP TO ESCAPE TO NEXT TEST

                ;SET UP TO LOOP WITH DATA          ; 3

013532 012777 004000 002116
013540 012767 000377 002210
013546 012777 000016 002102
013554 012777 177777 002104
013562 012777 015756 002074
013570 012777 033500 002064
013576 052777 000003 002056
013604 012777 040000 002056
013612 105777 002040
013616 100375
013620 017704 002034

                1$:  TSTB   #DHSCR
                BPL     1$
                MOV     #DHNRC,R4

                ;MASTER CLEAR INTERFACE
                ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
                ;SELECT LINE 16
                ;SET UP TO TRANSMIT 1 BYTE
                ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
                ;SET LINE SPEED FOR 9600 BAUD
                ;SET CHARACTER LENGTH FOR 10 BITS
                ;START TRANSMITTER
                ;WAIT TO RECEIVE CHARACTER

                ;(R4)=RECEIVED CHARACTER
                ;IN LOW BYTE, AND LINE NUMBER AND
                ;CHARACTER STATUS IN HIGH BYTE
                ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                ;AND LINE NUMBER AND CHARACTER
                ;STATUS IN HIGH BYTE
                ;ARE EXPECTED AND RECEIVED DATA THE SAME

013624 012705 107377

                MOV     #107377,R5

013630 020504
013632 001401
013634
013634 104000
013636 104400
000004
000011
000777
000017
100000
000000
000005
000037

                CMP      R5,R4
                BEQ      2$
                HLT
                EMT

                ;CHARACTER LENGTH, DATA

                ;OR LINE NUMBER ERROR

2$:   SCOPE
      LENGTX=LENGTX+1
      LENGTC=LENGTC+1
      XCODE=XCODE+2+1
      LINEX=LINEX+1
      BITX=BITX+BITX
      LENGTX=0
      LENGTC=5
      XCODE=37

```

```

000004      .REPT 4
            .NLIST
            DATAR=LINEX*400+100000+XCODE
            .LIST
            CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR
            .NLIST
            LENGTX=LENGTX+1
            LENGTC=LENGTC+1
            XCODE=XCODE+2+1
            .LIST
            .ENDR
013640 107437 DATAR=LINEX*400+100000+XCODE
            CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

            ;CHARACTER LENGTH TEST
            ;TRANSMIT 1 CHARACTER ON LINE 17
            ;CHARACTER LENGTH IS 5 BITS
            ;EXPECTED RECEIVED CHARACTER IS 37
            ;LINE SPEED IS 9600 BAUD

013640      TS \XN,400,2#
013640 012767 000340 164130 T75:  MOV #340,PS ;DISABLE ALL INTERRUPTS
013646 012767 000400 002050      MOV #400,ICOUNT ;SET UP FOR 400 ITERATIONS
013654 012767 013766 002036      MOV #2#,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

            .IF NB <>
            MOV #,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
            .ENDC
            XN=XN+1

013662 012777 004000 001766      MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
013670 012767 000037 002060      MOV #37,TDATA ;CHARACTER TO BE TRANSMITTED = 37(OCTAL)
013676 012777 000017 001752      MOV #17,@DHSCR ;SELECT LINE 17
013704 012777 177777 001754      MOV #-1,@DHBC ;SET UP TO TRANSMIT 1 BYTE
013712 012777 015756 001744      MOV #TDATA,@DHBA ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
013720 012777 033500 001734      MOV #33500,@DHLPR ;SET LINE SPEED FOR 9600 BAUD
013726 052777 000000 001726      BIS #0,@DHLPR ;SET CHARACTER LENGTH FOR 5 BITS
013734 012777 100000 001726      MOV #100000,@DHBAR ;START TRANSMITTER
013742 105777 001710      1#: TSTB @DHSCR ;WAIT TO RECEIVE CHARACTER
013746 100375      BPL 1#
013750 017704 001704      MOV @DHNRC,R4 ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
013754 012705 107437      MOV #107437,R5 ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
013760 020504      CMP R5,R4 ;ARE EXPECTED AND RECEIVED DATA THE SAME
013762 001401      BEQ 2#
013764      HLT ;CHARACTER LENGTH, DATA
013764 104000      EMT ;OR LINE NUMBER ERROR

013766 104400      2#: SCOPE
            LENGTX=LENGTX+1
            LENGTC=LENGTC+1
            XCODE=XCODE+2+1
            DATAR=LINEX*400+100000+XCODE
013770      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

            ;CHARACTER LENGTH TEST

```

```

;TRANSMIT 1 CHARACTER ON LINE 17
;CHARACTER LENGTH IS 6 BITS
;EXPECTED RECEIVED CHARACTER IS 77
;LINE SPEED IS 9600 BAUD

013770      TS \XN,400,2$
013770 012767 000340 164000 T76:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
013776 012767 000400 001720      MOV    #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
014004 012767 014116 001706      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                          .IF NB <>
                          MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                          .ENDC
                          XN=XN+1

014012      000077
014012 012777 004000 001636      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
014020 012767 000077 001730      MOV    #77,TDATA        ;CHARACTER TO BE TRANSMITTED = 77(OCTAL)
014026 012777 000017 001622      MOV    #17,@DHSCR       ;SELECT LINE 17
014034 012777 177777 001624      MOV    #-1,@DHBC        ;SET UP TO TRANSMIT 1 BYTE
014042 012777 015756 001614      MOV    #TDATA,@DHBA     ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
014050 012777 033500 001604      MOV    #33500,@DHLPR    ;SET LINE SPEED FOR 9600 BAUD
014056 052777 000001 001576      BIS    #1,@DHLPR        ;SET CHARACTER LENGTH FOR 6 BITS
014064 012777 100000 001576      MOV    #100000,@DHBAR   ;START TRANSMITTER
014072 105777 001560      1$:  TSTB   @DHSCR         ;WAIT TO RECEIVE CHARACTER
014076 100375
014100 017704 001554      BPL    1$
                          MOV    @DHNRC,R4          ;(R4)=RECEIVED CHARACTER
                                                    ;IN LOW BYTE, AND LINE NUMBER AND
                                                    ;CHARACTER STATUS IN HIGH BYTE
014104 012705 107477      MOV    #107477,R5       ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                                    ;AND LINE NUMBER AND CHARACTER
                                                    ;STATUS IN HIGH BYTE
014110 020504      CMP    R5,R4           ;ARE EXPECTED AND RECEIVED DATA THE SAME
014112 001401      BEQ    2$
014114      HLT
014114 104000      EMT
                                                    ;CHARACTER LENGTH, DATA
                                                    ;OR LINE NUMBER ERROR

014116 104400      2$:  SCOPE
014116 000002      LENGTX=LENGTX+1
014116 000007      LENGTC=LENGTC+1
014116 000177      XCODE=XCODE+2+1
014120 107577      DATAR=LINEX*400+100000+XCODE
014120      CHART1 \LINEX,\BITX,\LENGTX,\LENGTC,\XCODE,\DATAR

;CHARACTER LENGTH TEST
;TRANSMIT 1 CHARACTER ON LINE 17
;CHARACTER LENGTH IS 7 BITS
;EXPECTED RECEIVED CHARACTER IS 177
;LINE SPEED IS 9600 BAUD

014120      TS \XN,400,2$
014120 012767 000340 163650 T77:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
014126 012767 000400 001570      MOV    #400,ICOUNT       ;SET UP FOR 400 ITERATIONS
014134 012767 014246 001556      MOV    #2$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                          .IF NB <>
                          MOV    #,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                          .ENDC
                          XN=XN+1

014142      000100
014142 012777 004000 001506      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
014150 012767 000177 001600      MOV    #177,TDATA        ;CHARACTER TO BE TRANSMITTED = 177(OCTAL)

```

```

014156 012777 000017 001472      MOV      #17, @DHSCR      ;SELECT LINE 17
014164 012777 177777 001474      MOV      #-1, @DHBC      ;SET UP TO TRANSMIT 1 BYTE
014172 012777 015756 001464      MOV      @TDATA, @DHBA   ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
014200 012777 033500 001454      MOV      #33500, @DHLPR  ;SET LINE SPEED FOR 9600 BAUD
014206 052777 000002 001446      BIS      #2, @DHLPR      ;SET CHARACTER LENGTH FOR 7 BITS
014214 012777 100000 001446      MOV      #100000, @DHBAR ;START TRANSMITTER
014222 105777 001430      1$: TSTB   @DHSCR         ;WAIT TO RECEIVE CHARACTER
014226 100375      BPL      1$
014230 017704 001424      MOV      @DHNRC, R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
014234 012705 107577      MOV      #107577, R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
014240 020504      CMP      R5, R4
014242 001401      BEQ
014244      HLT
014244 104000      EMT      ;CHARACTER LENGTH, DATA
                                ;OR LINE NUMBER ERROR
014246 104400      2$: SCOPE
      000003      LENGTX=LENGTX+1
      000010      LENGTC=LENGTC+1
      000377      XCODE=XCODE+2+1
      107777      DATAR=LINEX+400+100000+XCODE
014250      CHART1 \LINEX, \BITX, \LENGTX, \LENGTC, \XCODE, \DATAR
                                ;CHARACTER LENGTH TEST
                                ;TRANSMIT 1 CHARACTER ON LINE 17
                                ;CHARACTER LENGTH IS 10 BITS
                                ;EXPECTED RECEIVED CHARACTER IS 377
                                ;LINE SPEED IS 9600 BAUD
014250      TS \XN, 400, 2$
014250 012767 000340 163520      T100: MOV      #340, PS      ;DISABLE ALL INTERRUPTS
014256 012767 000400 001440      MOV      #400, ICOUNT    ;SET UP FOR 400 ITERATIONS
014264 012767 014376 001426      MOV      #2$, ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <>
                                MOV      #, FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
014272 012777 004000 001356      MOV      @BIT11, @DHSCR  ;MASTER CLEAR INTERFACE
014300 012767 000377 001450      MOV      #377, TDATA     ;CHARACTER TO BE TRANSMITTED = 377(OCTAL)
014306 012777 000017 001342      MOV      #17, @DHSCR     ;SELECT LINE 17
014314 012777 177777 001344      MOV      #-1, @DHBC      ;SET UP TO TRANSMIT 1 BYTE
014322 012777 015756 001334      MOV      @TDATA, @DHBA   ;SET UP ADDRESS OF CHARACTER TO BE TRANSMITTED
014330 012777 033500 001324      MOV      #33500, @DHLPR  ;SET LINE SPEED FOR 9600 BAUD
014336 052777 000003 001316      BIS      #3, @DHLPR      ;SET CHARACTER LENGTH FOR 10 BITS
014344 012777 100000 001316      MOV      #100000, @DHBAR ;START TRANSMITTER
014352 105777 001300      1$: TSTB   @DHSCR         ;WAIT TO RECEIVE CHARACTER
014356 100375      BPL      1$
014360 017704 001274      MOV      @DHNRC, R4      ;(R4)=RECEIVED CHARACTER
                                ;IN LOW BYTE, AND LINE NUMBER AND
                                ;CHARACTER STATUS IN HIGH BYTE
014364 012705 107777      MOV      #107777, R5     ;(R5)=EXPECTED CHARACTER IN LOW BYTE
                                ;AND LINE NUMBER AND CHARACTER
                                ;STATUS IN HIGH BYTE
                                ;ARE EXPECTED AND RECEIVED DATA THE SAME
014370 020504      CMP      R5, R4

```

014372 001401
014374
014374 104000

BEQ 2#
HLT
EMT

;CHARACTER LENGTH, DATA

;OR LINE NUMBER ERROR

014376 104400
000004
000011
000777
000020
000000

2# : SCOPE
LENGTX=LENGTX+1
LENGTC=LENGTC+1
XCODE=XCODE+2+1
LINEX=LINEX+1
BITX=BITX+BITX

```

1 014400          .EOP      †/BEGIN/
                                ;END OF PASS
                                ;TYPE NAME OF TEST
                                ;UPDATE PASS COUNT
                                ;CHECK FOR EXIT TO ACT-11
                                ;RESTART TEST

014400  104401          EOP:    TYPE
014402  016362          MEPASS      ;TYPE NAME OF TEST
014404  005067  001344  CLR      LAST
014410  005067  001274  CLR      ERRFLG ;CLEAR LAST ERROR PC
014414  005267  001272  INC      PASCNT ;CLEAR ERROR FLAG
014420  005767  164356  TST      LIGHTS ;UPDATE PASS COUNT
014424  001005          BNE      2$    ; ARE WE USING LIGHTS?      : 4
014426  104401          TYPE          ; BRANCH IF WE ARE          : 6
014430  016375          PASTXT        ; TYPE PASCOUNT MESSAGE    : 5
014432  104402          OCTASC        ; PRINT PASSCOUNT        : 5
014434  014472          PASARG        ;                             : 4
014436  000403          BR      3$    ; CONTINUE                  : 4
014440          2$:
014440  016767  001246  164334  MOV      PASCNT,LIGHTS ;DISPLAY PASS COUNT        : 4
014446          3$:
014446  013701  000042  MOV      @42,R1
014452  001405          BEQ      RESTRT ;CHECK FOR ACT-11 OR DDP   : 4
014454  000005          RESET
014456  004711          LOGICAL:  JSR      PC,(R1) ;IF NOT, CONTINUE TESTING
014460  000240          NOP
014462  000240          NOP
014464  000240          NOP
014466  000167  164614  RESTRT:  JMP      BEGIN
014472  000001          PASARG:  .WORD   1      ; PARAMETERS TO PRINT PASCOUNT : 5
014474  006          .BYTE   6,2
014476  015712          .WORD   PASCNT
2 014500          .SCOPE
                                ;CHECK FOR LOOP ON CURRENT TEST
                                ;CHECK FOR ITERATION SUPPRESSION
                                : 3

014500  032777  002000  164272  SCOPER:  BIT      @SW10,@SWR
014506  001030          BNE      4$    ; 4
014510  032777  040000  164262  1$:   BIT      @SW14,@SWR
014516  001021          BNE      3$    ; 4
014520  032777  004000  164252  BIT      @SW11,@SWR
014526  001006          BNE      2$    ; 4
014530  005267  001172  INC      LPCNT
014534  026767  001166  001162  CMP      LPCNT,ICOUNT
014542  001007          BNE      3$
014544  005067  001156  2$:   CLR      LPCNT
014550  005067  001134  CLR      ERRFLG
014554  011667  001136  MOV      (SP),RETRN
014560  000002          RTI
014562  016716  001130  3$:   MOV      RETRN,(SP)
014566  000002          RTI
014570  005767  001114  4$:   TST      ERRFLG
014574  001745          BEQ      1$
014576  000762          BR      2$

```

3 014600

.SCOP1

;CHECK FOR FREEZE ON CURRENT DATA

014600	032777	001000	164172	SCOP1R:	BIT	@SW09,@SMR
014606	001402				BEQ	1*
014610	016716	001106			MOV	FREEZ1,(SP)
014614	000002			1*:	RTI	

: 4

1 014616

.ERROR

:ERROR HANDLER

```

014616 032777 020000 164154 ERRORS: BIT      #SW13,@SWR
014624 001055          BNE      HALTS
014626 021667 001122      CMP      (SP),LAST
014632 001404          BEQ      1#
014634 011667 001114      MOV      (SP),LAST
014640 005067 001044      CLR      ERRFLG
014644 104406          1#:    SAVOSP
014646 011605          MOV      (SP),R5
014650 162705 000002      SUB      #2,R5
014654 011504          MOV      (R5),R4
014656 006304          ASL      R4
014660 006304          ASL      R4
014662 042704 177001      BIC      #177001,R4
014666 062704 016514      ADD      @ERRTAB,R4
014672 012467 000040      MOV      (R4),ERRMSG
014676 011467 000052      MOV      (R4),DATABP
014702 005767 001002      TST      ERRFLG
014706 001403          BEQ      TYPMSG
014710 005767 000040      TST      DATABP
014714 001011          BNE      TYPDAT
014716 104401          TYPMSG: TYPE
014720 016272          MCRLF
014722 104402          OCTASC
014724 015022          ERTABO
014726 012767 000001 000754 MOV      #1,ERRFLG
014734 104401          TYPE
014736 000000          ERRMSG: 0
014740 005767 000010      TYPDAT: TST      DATABP
014744 001404          BEQ      RESREG
014746 104401          TYPE
014750 016272          MCRLF
014752 104402          OCTASC
014754 000000          DATABP: 0
014756 104407          RESREG: RES05
014760 005777 164014      HALTS:  TST      @SWR
014764 100005          BPL      EXITER
014766 010046          PUSHRO
014770 016600 000002      MOV      2(SP),R0
014774 000000          HALT
014776 012600          POPRO
015000 005267 000710      EXITER: INC      ERRCNT
015004 032777 002000 163766 BIT      #SW10,@SWR
015012 001402          BEQ      1#
015014 016716 000700      MOV      ESCAPE,(SP)
015020 000002          1#:    RTI
015022 000001          ERTABO: 1
015024 006          .BYTE 6.2
015026 015746          SAVPC

```

: 4

: 5
: 5
: 5

: 5
: 5

: 4

: 4

```

015030                                .TRPSRV
                                        ;TRAP DISPATCH SERVICE
                                        ;ARGUMENT OF TRAP IS EXTRACTED
                                        ;AND USED AS OFFSET TO OBTAIN POINTER
                                        ;TO SELECTED SUBROUTINE
                                        ; 3

015030 011646                                TRPSRV: MOV    (SP),-(SP)           ;GET PC OF RETURN
015032 162716 000002                        SUB     #2,(SP)           ;=PC OF TRAP
015036 017616 000000                        MOV     @8(SP),(SP)      ;GET TRP
015042 006316                                TRPOK: ASL   (SP)         ;MULTIPLY TRAP ARG BY 2
015044 042716 177001                        BIC    #177001,(SP)     ;CLEAR UNWANTED BITS
015050 062716 016434                        ADD     @TRPTAB,(SP)    ;POINTER TO SUBROUTINE ADDRESS
015054 017616 000000                        MOV     @8(SP),(SP)     ;SUBROUTINE ADDRESS
015060 000136                                JMP     @8(SP)          ;GO TO SUBROUTINE
2 015062                                .SAVREG

                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5
015062 016667 000004 000656 SV05P: MOV     4(SP),SAVPC

                                        ;SAVE R0-R5
015070 010567 000646                        SV05:  MOV    R5,SAVR5
015074 010467 000640                        MOV    R4,SAVR4
015100 010367 000632                        MOV    R3,SAVR3
015104 010267 000624                        MOV    R2,SAVR2
015110 010167 000616                        MOV    R1,SAVR1
015114 010067 000610                        MOV    R0,SAVR0
015120 000002                        RTI
3 015122                                .RESREG

                                        ;RESTORE R0-R5
015122 016700 000602                        RS05:  MOV    SAVR0,R0
015126 016701 000600                        MOV    SAVR1,R1
015132 016702 000576                        MOV    SAVR2,R2
015136 016703 000574                        MOV    SAVR3,R3
015142 016704 000572                        MOV    SAVR4,R4
015146 016705 000570                        MOV    SAVR5,R5
015152 000002                        RTI

```

1 015154

.TYPER

;TELETYPE OUTPUT ROUTINE

015154	017605	000000		TYPERS:	MOV	@(SP),R5
015160	062716	000002			ADD	@2,(SP)
015164	105777	000462		1\$:	TSTB	@TPCSR
015170	100375				BPL	1\$
015172	105715				TSTB	(R5)
015174	001001				BNE	2\$
015176	000002				RTI	
015200	112577	000450		2\$:	MOVB	(R5)+,@TPDBR
015204	000767				BR	1\$

; 3

2 015206

.INSTRG

;ASCII STRING INPUT ROUTINE

015206	017667	000000	000006	INSTRG:	MOV	@(SP),MSG
015214	062716	000002			ADD	@2,(SP)
015220	104401			INSTR1:	TYPE	
015222	000000			MSG:	0	
015224	012704	016456			MOV	@INBUF,R4
015230	012703	000007			MOV	@7,R3
015234	105777	000406		1\$:	TSTB	@TKCSR
015240	100375				BPL	1\$
015242	117714	000402			MOVB	@TKDBR,(R4)
015246	142714	000200			BICB	@200,(R4)
015252	122427	000015			CMPB	(R4)+,@15
015256	001413				BEQ	INSTR2
015260	117777	000364	000366		MOVB	@TKDBR,@TPDBR
015266	105777	000360		2\$:	TSTB	@TPCSR
015272	100375				BPL	2\$
015274	005303				DEC	R3
015276	001356				BNE	1\$
015300	104401			INSTRE:	TYPE	
015302	016266				MQM	
015304	000745				BR	INSTR1
015306	000002			INSTR2:	RTI	

1 015310

.PARAMS

;CONVERT ASCII STRING TO OCTAL

; 3

015310 011605
 015312 012567 000146
 015316 012567 000144
 015322 012567 000142
 015326 112567 000140
 015332 112567 000135
 015336 010516
 015340 005005
 015342 012704 016456
 015346 122714 000015
 015352 001420
 015354 121427 000060
 015360 002415
 015362 121427 000067
 015366 003012
 015370 142714 000060
 015374 152405
 015376 122714 000015
 015402 001406
 015404 006305
 015406 006305
 015410 006305
 015412 000760
 015414 104404
 015416 000750

PARAMS: MOV (SP),R5
 MOV (R5)+,LOLIM
 MOV (R5)+,HILIM
 MOV (R5)+,DEVADR
 MOV (R5)+,LOBITS
 MOV (R5)+,ADRCNT
 MOV R5,(SP)
 PARAM1: CLR R5
 MOV #INBUF,R4
 CMPB #15,(R4)
 BEQ PARERR
 1\$: CMPB (R4),#60
 BLT PARERR
 CMPB (R4),#67
 BGT PARERR
 BICB #60,(R4)
 BISB (R4)+,R5
 CMPB #15,(R4)
 BEQ LIMITS
 ASL R5
 ASL R5
 ASL R5
 BR 1\$
 PARERR: INSTER
 BR PARAM1

;TEST TO SEE IF NUMBER IS WITHIN LIMITS

015420 020567 000042
 015424 101373
 015426 020567 000032
 015432 103770
 015434 136705 000032
 015440 001365

LIMITS: CMP R5,HILIM
 BHI PARERR
 CMP R5,LOLIM
 BLO PARERR
 BITB LOBITS,R5
 BNE PARERR

; 3

;STORE NUMBER AT SPECIFIED ADDRESS

015442 016704 000022
 015446 010524
 015450 062705 000002
 015454 105367 000013
 015460 001372
 015462 000002
 015464 000000
 015466 000000
 015470 000000
 015472 000000
 015473

1\$: MOV DEVADR,R4
 MOV R5,(R4)+
 ADD #2,R5
 DECB ADRCNT
 BNE 1\$
 RTI
 LOLIM: 0
 HILIM: 0
 DEVADR: 0
 LOBITS: 0
 ADRCNT=LOBITS+1

015474

.OCTASC

;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

015474	017601	000000	OCTASN: MOV	@(SP),R1	
015500	062716	000002		ADD	#2,(SP)
015504	012167	000130		MOV	(R1)+,WRDCNT
015510	112167	000126	1\$:	MOVB	(R1)+,CHRCNT
015514	112167	000123		MOVB	(R1)+,SPACNT
015520	013167	000120		MOV	@(R1)+,BINWRD
015524	016704	000114	2\$:	MOV	BINWRD,R4
015530	116705	000106		MOVB	CHRCNT,R5
015534	012700	016470		MOV	#TEMP,R0
015540	010403		3\$:	MOV	R4,R3
015542	042703	177770		BIC	#177770,R3
015546	062703	000260		ADD	#260,R3
015552	110320			MOVB	R3,(R0)+
015554	006204			ASR	R4
015556	006204			ASR	R4
015560	006204			ASR	R4
015562	005305			DEC	R5
015564	001365			BNE	3\$
015566	012703	016502		MOV	#MDATA,R3
015572	114023		4\$:	MOVB	-(R0),(R3)+
015574	105367	000042		DECB	CHRCNT
015600	001374			BNE	4\$
015602	105767	000035		TSTB	SPACNT
015606	001405			BEQ	6\$
015610	112723	000240	5\$:	MOVB	#240,(R3)+
015614	105367	000023		DECB	SPACNT
015620	001373			BNE	5\$
015622	105013		6\$:	CLRB	(R3)
015624	104401			TYPE	
015626	016502			MDATA	
015630	005367	000004		DEC	WRDCNT
015634	001325			BNE	1\$
015636	000002			RTI	
015640	000000		WRDCNT: 0		
015642	000000		CHRCNT: 0		
	015643		SPACNT=CHRCNT+1		
015644	000000		BINWRD: 0		

015646

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

015646 177560
015650 177562
015652 177564
015654 177566

TKCSR: 177560
TKDBR: 177562
TPCSR: 177564
TPDBR: 177566
.IRP A

TLVL>

<DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DH

015656 000000
015660 000000
015662 000000
015664 000000
015666 000000
015670 000000
015672 000000
015674 000000
015676 000000
015700 000000
015702 000000
015704 000000
015706 000000
2 015710

A: 0
.ENDM
DHSCR: 0
DHNRC: 0
DHLPR: 0
DHBA: 0
DHBC: 0
DHBAR: 0
DHBCR: 0
DHSSR: 0
DHSLR: 0
DHRVEC: 0
DHRLVL: 0
DHTVEC: 0
DHTLVL: 0

.VARIA <+/TDATA/>
;PROGRAM VARIABLES

015710 000000
015712 000000
015714 000000
015716 000000
015720 000000
015722 000000
015724 000000
015726 000000
015730 000000
015732 000000
015734 000000
015736 000000
015740 000000
015742 000000
015744 000000
015746 000000
015750 000000
015752 000000
015754 000000

ERRFLG: 0 ;ERROR FLAG
PASCNT: 0 ;PASS COUNT
ERRCNT: 0 ;ERROR COUNT
RETRN: 0 ;SCOPE RETURN ADDRESS FOR TEST LOOPING
ESCAPE: 0 ;ADDRESS FOR ERROR ESCAPE
FREEZ1: 0 ;DATA LOOPING RETURN ADDRESS
ICOUNT: 0 ;ITERATION COUNT FOR TEST IN PROGRESS
LPCNT: 0 ;NUMBER OF ITERATIONS THIS TEST
SAVR0: 0 ;R0 SAVE AREA
SAVR1: 0 ;R1 SAVE AREA
SAVR2: 0 ;R2 SAVE AREA
SAVR3: 0 ;R3 SAVE ARE
SAVR4: 0 ;R4 SAVE AREA
SAVR5: 0 ;R5 SAVE AREA
SAVSP: 0 ;STACK POINTER SAVE AREA
SAVPC: 0 ;CALLING ROUTINE SAVE AREA
INIFLG: 0 ;PROGRAM INITIALIZATION FLAG
STFLG: 0 ;PROGRAM START FLAG
LAST: 0 ;LAST ERROR PC

.IRP A <+/TDATA/>
A: 0
.ENDM
TDATA: 0

015756 000000

; 3

1 015760

.PFAIL

;ENTER HERE ON POWER FAILURE

```

015760 010046          PFAIL:  MOV    R0,-(SP)          ;SAVE R0-R5 ON PROCESSOR STACK
015762 010146          MOV    R1,-(SP)
015764 010246          MOV    R2,-(SP)
015766 010346          MOV    R3,-(SP)
015770 010446          MOV    R4,-(SP)
015772 010546          MOV    R5,-(SP)
015774 016746 162024   MOV    24,-(SP)
016000 010667 177740   MOV    SP,SAVSP          ;SAVE STACK POINTER
016004 012767 016016 162012 MOV    #RESTART,24      ;SET UP FOR POWER UP TRAP
016012 000000          HALT                                ;HALT ON POWER DOWN NORMAL
016014 000777          BR

```

;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

```

016016 016706 177722   RESTAR: MOV    SAVSP,SP          ;RESTORE STACK POINTER
016022 012605          MOV    (SP)+,R5          ;RESTORE R0-R5
016024 012604          MOV    (SP)+,R4
016026 012603          MOV    (SP)+,R3
016030 012602          MOV    (SP)+,R2
016032 012601          MOV    (SP)+,R1
016034 012600          MOV    (SP)+,R0
016036 012767 015760 161760 MOV    #PFAIL,24        ;SET UP FOR POWER FAILURE
016044 012767 000340 161724 MOV    #340,PS
016052 012706 017004   MOV    #STACK,SP
016056 005067 000406   CLR    TEMP
016062 005267 000402   INC    TEMP
016066 001375          BNE    .-4
016070 104401          TYPE
016072 016272          MCRLF
016074 104402          OCTASC
016076 016120          PFTAB
016100 104401          TYPE
016102 016275          MPFAIL
016104 005067 177600   CLR    ERRFLG
016110 005067 177640   CLR    LAST
016114 000177 177576   JMP    #RETRN
016120 000001          PFTAB: 1
016122 000006 000002   6,2
016126 015716          RETRN

```

; 5
; 5

016130				.MSG	↑/DH11 CHARACTER LENGTH AND BASIC DATA TEST/↑/CZDHE-CO/
016130	015	012	012	MTITLE:	.ASCIZ <15><12><12>/DH11 CHARACTER LENGTH AND BASIC DATA TEST /<15><12>
016133	104	110	061		
016136	061	040	103		
016141	110	101	122		
016144	101	103	124		
016147	105	122	040		
016152	114	105	116		
016155	107	124	110		
016160	040	101	116		
016163	104	040	102		
016166	101	123	111		
016171	103	040	104		
016174	101	124	101		
016177	040	124	105		
016202	123	124	040		
016205	015	012	000		
016210	015	012	126	MVECTOR:	.ASCIZ <15><12>/VECTOR ADDRESS-/
016213	105	103	124		
016216	117	122	040		
016221	101	104	104		
016224	122	105	123		
016227	123	055	000		
016232	015	012	103	MREGAD:	.ASCIZ <15><12>/CONTROL REGISTER ADDRESS-/
016235	117	116	124		
016240	122	117	114		
016243	040	122	105		
016246	107	111	123		
016251	124	105	122		
016254	040	101	104		
016257	104	122	105		
016262	123	123	055		
016265	000				
016266	040	040	077	MQM:	.ASCIZ / ?/
016271	000				
016272	015	012	000	MCRLF:	.ASCIZ <15><12>
016275	040	040	120	MPFAIL:	.ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
016300	117	127	105		
016303	122	040	106		
016306	101	111	114		
016311	125	122	105		
016314	054	040	120		
016317	122	117	107		
016322	122	101	115		
016325	040	122	105		
016330	123	124	101		
016333	122	124	040		
016336	101	124	040		
016341	124	105	123		
016344	124	040	111		
016347	116	040	120		
016352	122	117	107		
016355	122	105	123		
016360	123	000			
016362	015	012	103	MEPASS:	.ASCIZ <15><12>/CZDHE-CO/
016365	132	104	110		

```

016370      105      055      103
016373      060      000
016375      015      012      120 PASTXT: .ASCIZ <15><12>/PASS COUNT = /
016400      101      123      123
016403      040      103      117
016406      125      116      124
016411      040      075      040
016414      000
016415      015      012      122 MR:      .ASCIZ <15><12>/R/
016420      000
016421      015      012      124 MTSTPC: .ASCIZ <15><12>/TEST PC-/
016424      105      123      124
016427      040      120      103
016432      055      000

2
3 016434      .EVEN
                .EVEN
                .TRPTAB

                ;TABLE OF POINTERS FOR TRAP DECODING

016434      014500      TRPTAB: SCOPER
016436      015154      TYPER
016440      015474      OCTASN
016442      015206      INSTRG
016444      015300      INSTRE
016446      015310      PARAMS
016450      015062      SV05P
016452      015122      RS05
016454      014600      SCOP1R
4 016456      .BUFFER

                ;BUFFERS FOR INPUT-OUTPUT

016456      000000      INBUF: 0
                016470      . = .+10
016470      000000      TEMP: 0
                016502      . = .+10

016502      000000      MDATA: 0
                016514      . = .+10
5 016514      .ERRTAB

                ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA

016514      ERRTAB:
6 016514      016520      EM1
7 016516      016572      DT1
8 016520      103      110      101      EM1: .ASCIZ /CHARACTER LENGTH ERROR/<15><12>/EXP REC /
016523      122      101      103
016526      124      105      122
016531      040      114      105
016534      116      107      124
016537      110      040      105
016542      122      122      117
016545      122      015      012
016550      105      130      120
016553      040      040      040

```

	016556	040	040	122	
	016561	105	103	040	
	016564	040	040	040	
	016567	040	000		
9					.EVEN
10	016572	000002			DT1: 2
11	016574	006	002		.BYTE 6,2
12	016576	015742			SAVR5
13	016600	006	000		.BYTE 6,0
14	016602	015740			SAVR4
15	016604				.ENDCOD
	016604	000000			ENDCOD: 0
16		000001			.END

ADRCNT = 015473	ERRTAB 016514	PFAIL 015760	SW11 = 004000	T4 002010
BEGIN 001306	ERTABO 015022	PFTAB 016120	SW12 = 010000	T40 006650
BINWRD 015644	ESCAPE 015720	POPRO = 012600	SW13 = 020000	T41 007000
BITX = 000000	EXITER 015000	POP1SP= 005726	SW14 = 040000	T42 007130
BIT00 = 000001	FREEZ1 015722	POP2SP= 022626	SW15 = 100000	T43 007260
BIT01 = 000002	HALTS 014760	PS = 177776	TDATA 015756	T44 007410
BIT02 = 000004	HILIM 015466	PUSHRO= 010046	TEMP 016470	T45 007540
BIT03 = 000010	ICOUNT 015724	PUSH1S= 005746	TKCSR 015646	T46 007670
BIT04 = 000020	INBUF 016456	PUSH2S= 024646	TKDBR 015650	T47 010020
BIT05 = 000040	INIFLG 015750	RDATA = 107777	TPCSR 015652	T5 002140
BIT06 = 000100	INSTER= 104404	RESREG 014756	TPDBR 015654	T50 010150
BIT07 = 000200	INSTR = 104403	RESTAR 016016	TRPCK 015042	T51 010300
BIT08 = 000400	INSTRE 015300	RESTRT 014466	TRPSRV 015030	T52 010430
BIT09 = 001000	INSTRG 015206	RESO5 = 104407	TRPTAB 016434	T53 010560
BIT10 = 002000	INSTR1 015220	RETRN 015716	TYPDAT 014740	T54 010710
BIT11 = 004000	INSTR2 015306	RSO5 015122	TYPE = 104401	T55 011040
BIT12 = 010000	LAST 015754	SAVPC 015746	TYPER 015154	T56 011170
BIT13 = 020000	LENGTC= 000011	SAVRO 015730	TYPMSG 014716	T57 011320
BIT14 = 040000	LENGTX= 000004	SAVR1 015732	T1 001400	T6 002270
BIT15 = 100000	LIGHTS 001002	SAVR2 015734	T10 002550	T60 011450
CHRCNT 015642	LIMITS 015420	SAVR3 015736	T100 014250	T61 011600
CLENGT= 000011	LINEX = 000020	SAVR4 015740	T11 002700	T62 011730
CODEX = 000777	LOBITS 015472	SAVR5 015742	T12 003030	T63 012060
DATABP 014754	LOGICA 014456	SAVSP 015744	T13 003160	T64 012210
DATAR = 107777	LOLIM 015464	SAVO5P= 104406	T14 003310	T65 012340
DEVADR 015470	LPCNT 015726	SCOPE = 104400	T15 003440	T66 012470
DHBA 015664	MCRLF 016272	SCOPE1= 104410	T16 003570	T67 012620
DHBAR 015670	MATA 016502	SCOPE1R 014600	T17 003720	T7 002420
DHBC 015666	MEPASS 016362	SPACNT= 015643	T2 001530	T70 012750
DHBCR 015672	MPFAIL 016275	STACK = 017004	T20 004050	T71 013100
DHLPR 015662	MQM 016266	START 001004	T21 004200	T72 013230
DHNRC 015660	MR 016415	STFLG 015752	T22 004330	T73 013360
DHRLVL 015702	MREGAD 016232	SV05 015070	T23 004460	T74 013510
DHRVEC 015700	MSG 015222	SV05P 015062	T24 004610	T75 013640
DHSCR 015656	MTITLE 016130	SWR 001000	T25 004740	T76 013770
DHSLR 015676	MTSTPC 016421	SW00 = 000001	T26 005070	T77 014120
DHSSR 015674	MVECTO 016210	SW01 = 000002	T27 005220	VEC1 001164
DHTLVL 015706	N = 000001	SW02 = 000004	T3 001660	VEC2 001174
DHTVEC 015704	OCTASC= 104402	SW03 = 000010	T30 005350	WRDCNT 015640
DT1 016572	OCTASN 015474	SW04 = 000020	T31 005500	X = 000000
EM1 016520	PARAM = 104405	SW05 = 000040	T32 005630	XBIT = 000000
ENDCOD 016604	PARAMS 015310	SW06 = 000100	T33 005760	XCODE = 000777
EOP 014400	PARAM1 015340	SW08 = 000400	T34 006110	XLNGT= 000004
ERRCNT 015714	PARERR 015414	SW09 = 001000	T35 006240	XLIN = 000020
ERRFLG 015710	PASARG 014472	SW10 = 002000	T36 006370	XN = 000101
ERRMSG 014736	PASCNT 015712		T37 006520	Y = 000011
ERRORS 014616	PASTXT 016375			

. ABS. 016606 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 17664 WORDS (69 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
CZDHEC.BIN,CZDHEC.SEQ=CZDHEC.DOC,DHMACA.MAC,CZDHEC.P11