

DH11

DH11 SINGLE LINE DATA
CZDHFC0

AH-FG25C-MC
1 OF 1 OCT 1985
COPYRIGHT © 1972-85

digital
MADE IN USA

LINE NO.	LINE DATA	LINE DATA	LINE DATA	LINE DATA	LINE DATA	LINE DATA
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



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

.REM !

IDENTIFICATION

PRODUCT CODE: AC-8464C-MC
PRODUCT NAME: CZDHFCO DH11 SINGLE LINE 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

1. ABSTRACT

THE DH11 SINGLE LINE DATA TEST VERIFIES THAT ALL CHARACTERS (0-377), EACH LINE CAN TRANSMIT AND RECEIVE AT ALL SPEEDS (8 BITS PER CHARACTER) AND ALL CHARACTER LENGTHS (AT A SPEED OF 9600 BAUD).

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 SINGLE LINE 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 SINGLE LINE 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 SINGLE LINE 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 ENTERD 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 R0. 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 "CZDHF-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 "CZDHF-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 TYPE ON SUCCEEDING ERROR TYPEOUTS

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

6.1.1 ERROR DESCRIPTIONS

SEE LISTING FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15=0

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

6.2.2 SW15=1

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

6.2.3 ILLEGAL INTERRUPTS

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

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

- 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 CZDHF-C TO END OF TYPEOUT OF CZDHF-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
20
21
22
23
24
25
26

9. PROGRAM DESCRIPTION

THE FIRST GROUP OF TESTS TRANSMITS ALL CHARACTERS (0-377) ONE AT A TIME AT A LINE SPEED OF 9600 BAUD AND A CHARACTER LENGTH OF 8 BITS. EACH LINE IS CHECKED IN AN INDIVIDUAL LOOP. EACH TEST IN THIS GROUP CAN BE SET UP UNDER PROGRAM CONTROL TO LOOP ON A SINGLE CHARACTER USING THE FREEZE ON DATA (SW09) OPTION.

THE NEXT GROUP OF TESTS VERIFIES THAT ALL CHARACTERS CAN BE TRANSMITTED AT EACH STANDARD SPEED (50-9600 BAUD) AT 8 BITS PER CHARACTER. EACH LINE IS TESTED IN AN INDIVIDUAL TEST LOOP, AND A PARTICULAR SPEED CAN BE CHOSEN FOR SCOPING BY USING THE FREEZE ON DATA OPTION.

THE FINAL GROUP OF TESTS TRANSMITS ALL CHARACTERS AT EACH CHARACTER LENGTH (5-8 BITS) AT 9600 BAUD ON A SINGLE LINE. EACH LINE IS TESTED IN AN INDIVIDUAL TEST LOOP AND A PARTICULAR CHARACTER LENGTH CAN BE CHOSEN FOR SCOPING USING THE FREEZE ON DATA (SW09) OPTION.

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

2
3
5 000000

.LIST ME
.NLIST MC,MD,CND
.HEADER †/1972, 1976, 1985/,†/DH11 SINGLE LINE DATA TEST/,†/CZDHF-CO/

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE DH11 SINGLE LINE 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 " CZDHF-CO "
;AND THEN RESUM TESTING

; 3

000000
6 000000

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

;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
G20164      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 ; 3
020000      BIT13=20000
010000      BIT12=10000
004000      BIT11=4000
002000      BIT10=2000
001000      BIT09=1000
000400      BIT08=400
000200      BIT07=200
000100      BIT06=100
000040      BIT05=40
000020      BIT04=20
000010      BIT03=10
000004      BIT02=4
000002      BIT01=2
000001      BIT00=1
1 000000    .CATCH

```


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

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

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

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

```

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

1 000204    .TRPDEF

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

000204    TRPDEF  SCOPE,+ /SCOPE LOOP AND ITERATION HANDLER/
          104400    SCOPE=TRAP+Y          ;SCOPE LOOP AND ITERATION HANDLER
          000001    Y=Y+1

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

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

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

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

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

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

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

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

2          . =46
3 000046    LOGICAL
4          . =52
5 000052    40000
6          .M^CRO
7          CODEM1
8          MOV      DHSSR,DHSLR          ;SET UP ADDRESS OF SILO
9          INC      DHSLR              ;STATUS REGISTER HIGH BYTE
10 000054    .ENDM CODEM1
          .START DHRVEC,3,4,DHSCR,0,177776,7,10...1

```

```

0          001000          . =1000

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

001000 177570          SWR: .WORD 177570          ; SWITCH DHSCR ADDRESS          ; 4
001002 177570          LIGHTS: .WORD 177570      ; LIGHTS                          ; 4
                                                ; 4

001004 012767 000340 176764 START: MOV #340,PS          ;LOCK OUT INTERRUPTS
001012 012706 020164          MOV #STACK,SP          ;SET UP PROCESSOR STACK
001016 012702 000024          MOV #24,R2             ; POINT TO VECTOR AREA          ; 7
001022 012722 016764          MOV #PFail,(R2)+        ;SET UP POWER FAIL TRAP        ; 7
001026 012722 000340          MOV #340,(R2)+        ;SERVICE AT LEVEL 7          ; 7
001032 012722 015216          MOV #ERRORS,(R2)+      ;ERROR HANDLER                  ; 7
001036 012722 000340          MOV #340,(R2)+        ;SERVICE AT LEVEL 7          ; 7
001042 012722 015430          MOV #TRPSRV,(R2)+      ;GENERAL HANDLER DISPATCH SERVICE ; 7
001046 012712 000340          MOV #340,(R2)          ;SERVICE AT LEVEL 7          ; 8
001052 005067 015274          CLR STFLG          ;CLEAR TEST START FLAG
001056 005067 015230          CLR PASCNT          ;CLEAR PASS COUNT
001062 005067 015226          CLR ERRCNT          ;CLEAR ERROR COUNT
001066 005067 015216          CLR ERRFLG          ;CLEAR ERROR FLAG
001072 005067 015212          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$:
001122 012767 000176 177650 MOV #176,SWR          ; POINT TO SOFT SWITCH DHSCR    ; 4
001130 005067 177646          CLR LIGHTS          ; 0 MEANS WE ARE NOT GOING TO USE LIGHTS ; 4
001134          2$:
001134 005726          TST (SP)+          ; CLEAN UP STACK                ; 5
001136 005726          TST (SP)+          ; 4
001140 012667 176642          MOV (SP)+,6        ; 4
001144 012667 176634          MOV (SP)+,4        ; 4
001150 104401 017134          TYPE .MTITLE          ;TYPE TITLE MESSAGE
001154 005767 015170          TST INIFLG          ;CHECK INITIALIZATION FLAG
                                .IF NB <DHRVEC>
001160 001001          BNE VEC1          ;IF NOT 0, CHECK SWITCHES
                                ;FOR REINITIALIZATION
                                .IFF
                                BNE BEGIN          ;IF NOT 0, START TEST
                                .ENDC
                                .IF NB <>
                                SIZE: CLR R0
                                MOV #2$,@#4
                                1$: TST (R0)+          ;SET UP TIME OUT RETURN
                                BR 1$          ;WILL TRAP WHEN NO MEMORY      ; 9
                                2$: MOV R0,HCORE          ;LOCATION RESPONDED, CONTINUE
                                SUB #2,HCORE          ;R0 CONTAINS ADDRESS OF
                                MOV #6,@#4          ;NON EXISTANT MEMORY          ; 9
                                ;RESTORE TRAPCATCHER

```

```

.ENDC
.IF NB <>
TRACER: MOV    #1$,#010           ;SET UP ILLEGAL INSTRUCTION TRAP RETURN
        SXT    R0                ;DO 11/40, 11/45 INSTRUCTION
        MOV    #RTT,TRTRET       ;11/40,45 RTT RETURN FROM TRACE TRAP
        BR     2$
1$:     MOV    #RTI,TRTRET       ;1105,10,20 RTI RETURN FROM TRACE TRAP
        MOV    #12,#010         ;RESTROE TRAPCATCHER
        MOV    #TRTRET,#016     ;SET UP TRACE TRAP VECTOR

.ENDC
.IF NB <DHRVEC>
.IF B   <>
001162 000404 BR     VEC2
.IFF
        TST    INIFLG           ;IF INITIALIZE FLAG=0
        BEQ    VEC2           ;GET VECTOR AND CSR ADDRESS

.ENDC
001164 032777 000001 177606 VEC1: BIT    #SW00,#SWR       ;IF SW00=1, GET NEW VECTOR ; 4
001172 001445 BEQ    BEGIN      ;AND CSR
001174 VEC2:
001174 012701 000300 MOV    #300,R1 ; 4
001200 012702 000302 MOV    #302,R2 ; 4
001204 012703 000004 MOV    #4,R3
001210 010211 1$: MOV    R2,(R1) ;RESTORE TRAPCATCHER
001212 005012 CLR    (R2) ;IN FLOATING VECTOR AREA
001214 060301 ADD    R3,R1
001216 060302 ADD    R3,R2
001220 020127 001000 CMP    R1,#1000
001224 001371 BNE    1$
001226 104403 INSTR ;INPUT ADDRESS OF DEVICE VECTOR
001230 017175 MVECTOR ;MESSAGE "VECTOR ADDRESS "
001232 104405 PARAM ;CONVERT STRING TO OCTAL
001234 000300 300 ;LOW LIMIT
001236 000770 770 ;HIGH LIMIT ; 3
001240 016300 DHRVEC ;LOCATIONS TO BE FILLED
001242 003 .BYTE 3 ;NUMBER OF LOCATIONS
001243 004 .BYTE 4 ;LSB MASK
001244 104403 INSTR ;INPUT ADDRESS OF DEVICE CSR
001246 017217 MREGAD ;MESSAGE "CONTROL REGISTER ADDRESS-"
001250 104405 PARAM ;CONVERT STRING TO OCTAL
001252 000000 0 ;LOW LIMIT
001254 177776 177776 ;HIGH LIMIT
001256 016256 DHSCR ;LOCATIONS TO BE FILLED
001260 007 .BYTE 7 ;NUMBER OF LOCATIONS
001261 010 .BYTE 10 ;LSB MASK

.ENDC
.IF NB <1>
001262 CODEM1
001262 016767 015006 015006 MOV    DHSSR,DHSLR ;SET UP ADDRESS OF SILO
001270 005267 015002 INC    DHSLR ;STATUS REGISTER HIGH BYTE

.ENDC
001274 005767 015050 TST    INIFLG ;IF INITIALIZATION FLAG
001300 001002 BNE    BEGIN ;IS CLEARED
001302 005167 015042 COM    INIFLG ;SET IT

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

```

```

001306 012767 000340 176462 BEGIN: MOV    #340,PS          ;LOCK OUT INTERRUPTS
001314 012706 020164          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 017406          MTSTPC          ;MESSAGE "TEST PC"
001334 104405          PARAM          ;CONVERT STRING TO OCTAL
001336 000000          0
001340 017500          17500
001342 016316          RETRN
001344    001          .BYTE 1
001345    001          .BYTE 1
001346 000410          BR     2$
001350 012767 001400 014740 1$:  MOV    #T1,RETRN      ;NORMAL START, TEST 1
001356 005767 014770          TST    STFLG          ;IF LOOPING, BYPASS TYPEOUT
001362 001004          BNE    3$
001364 005167 014762          COM    STFLG
001370 104401 017402          2$:  TYPE  ,MR
001374 000177 014716          3$:  JMP    @RETRN        ;TYPE "R" TO INDICATE START ; 3
;START TESTING

```



```

1
4      000000      LINE=0
5      000000      XLINE=LINE
6      000001      BITX=1
7      000001      XBIT=BITX
9      000020      .REPT 20
10     SDATA1 \LINE,\BITX
11     .NLIST
12     LINE=LINE+1
13     BITX=BITX+BITX
14     .LIST
15     .ENDR
001400 SDATA1 \LINE,\BITX

;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 0.
;CHARACTER LENGTH IS 8 BITS.
;LINE SPEED IS 9600 BAUD.

001400 TS \XN,10,4$,1$
001400 012767 000340 176370 T1:  MOV  #340,PS ;DISABLE ALL INTERRUPTS
001406 012767 000010 014710      MOV  #10,ICOUNT ;SET UP FOR 10 ITERATIONS
001414 012767 001540 014676      MOV  #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

001422 012767 001464 014672      .IF NB <1$>
      MOV  #1$,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
      .ENDC
      XN=XN+1

001430 012777 004000 014620      MOV  #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
001436 012703 000000      MOV  #0,R3 ;SET UP LINE NUMBER
001442 012767 100000 014706      MOV  #0*400+100000,TDATA ;SET EXPECTED LINE NUMBER
;AND VALID DATA FLAG
;EXPECTED DATA
;SELECT LINE 0
;SELECT 8 BITS CHARACTER
;LENGTH, 9600 BAUD SPEED
;FOR LINE 0

001450 012777 000000 014600      MOV  #0,@DHSCR
001456 012777 033503 014576      MOV  #33503,@DHLPR ;SELECT 8 BITS CHARACTER
;LENGTH, 9600 BAUD SPEED
;FOR LINE 0

001464 012777 177777 014574 1$:  MOV  #-1,@DHBC ;TRANSMIT 1 CHARACTER
001472 012777 016356 014564      MOV  #TDATA,@DHBA ;ADDRESS OF TRANSMIT DATA
001500 012777 000001 014562      MOV  #1,@DHBAR ;START TRANSMITTER
001506 105777 014544      2$:  TSTB @DHSCR ;WAIT FOR CHARACTER
001512 100375      BPL 2$ ;TO BE RECEIVED
001514 017704 014540      MOV  @DHNRC,R4 ;GET RECEIVED CHARACTER
001520 020467 014632      CMP  R4,TDATA ;COMPARE EXPECTED AND
001524 001401      BEQ 3$ ;RECEIVED DATA
001526      HLT 0 ;DATA ERROR
001526 104000      EMT 0
001530 104410      3$:  SCOPE1 ;CHECK FOR LOOP WITH CURRENT DATA
001532 105267 014620      INCB TDATA ;UPDATE TRANSMIT DATA
001536 001352      BNE 1$
001540 104400      4$:  SCOPE ;CHECK FOR ITERATIONS, LOOP
      000001
      000002
001542 SDATA1 \LINE,\BITX

;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 1.
;CHARACTER LENGTH IS 8 BITS.
;LINE SPEED IS 9600 BAUD.

```

```

001542      TS \XN,10,4$,1$
001542 012767 000340 176226 T2:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
001550 012767 000010 014546      MOV    #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
001556 012767 001702 014534      MOV    #4$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST

001564 012767 001626 014530      .IF NB <1$>
                                MOV    #1$,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

001572 012777 004000 014456      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
001600 012703 000001 014544      MOV    #1,R3             ;SET UP LINE NUMBER
001604 012767 100400 014544      MOV    #1*400+100000,TDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;SELECT LINE 1
                                ;SELECT 8 BITS CHARATER
                                ;LENGTH, 9600 BAUD SPEED
                                ;FOR LINE 1
                                ;TRANSMIT 1 CHARACTER
                                ;ADDRESS OF TRANSMIT DATA
                                ;START TRANSMITTER
                                ;WAIT FOR CHARACTER
                                ;TO BE RECEIVED
                                ;GET RECEIVED CHARACTER
                                ;COMPARE EXPECTED AND
                                ;RECEIVED DATA
                                ;DATA ERROR

001612 012777 000001 014436      MOV    #1,@DHSCR
001620 012777 033503 014434      MOV    #33503,@DHLPR

001626 012777 177777 014432 1$:  MOV    #-1,@DHBC
001634 012777 016356 014422      MOV    #TDATA,@DHBA
001642 012777 000002 014420      MOV    #2,@DHBAR
001650 105777 014402 2$:  TSTB  @DHSCR
001654 100375      BPL    2$
                                ;WAIT FOR CHARACTER
                                ;TO BE RECEIVED
001656 017704 014376      MOV    @DHNR,R4
                                ;GET RECEIVED CHARACTER
001662 020467 014470      CMP    R4,TDATA
                                ;COMPARE EXPECTED AND
001666 001401      BEQ    3$
                                ;RECEIVED DATA
001670      HLT    0
                                ;DATA ERROR
001670 104000      EMT    0
001672 104410 3$:  SCOPE1
001674 105267 014456      INCB  TDATA
                                ;CHECK FOR LOOP WITH CURRENT DATA
                                ;UPDATE TRANSMIT DATA
001700 001352      BNE    1$
001702 104400 4$:  SCOPE
                                ;CHECK FOR ITERATIONS. LOOP
                                000002
                                LINE=LINE+1
                                BITX=BITX+BITX
001704 000004      SDATA1 \LINE,\BITX

                                ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 2.
                                ;CHARACTER LENGTH IS 8 BITS.
                                ;LINE SPEED IS 9600 BAUD.

001704      TS \XN,10,4$,1$
001704 012767 000340 176064 T3:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
001712 012767 000010 014404      MOV    #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
001720 012767 002044 014372      MOV    #4$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST

001726 012767 001770 014366      .IF NB <1$>
                                MOV    #1$,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1

001734 012777 004000 014314      MOV    #BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
001742 012703 000002 014402      MOV    #2,R3             ;SET UP LINE NUMBER
001746 012767 101000 014402      MOV    #2*400+100000,TDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;SELECT LINE 2
                                ;SELECT 8 BITS CHARATER

001754 012777 000002 014274      MOV    #2,@DHSCR
001762 012777 033503 014272      MOV    #33503,@DHLPR

```

```

;LENGTH, 9600 BAUD SPEED
;FOR LINE 2
001770 012777 177777 014270 1$: MOV # -1, @DHBC ;TRANSMIT 1 CHARACTER
001776 012777 016356 014260 MOV @TDATA, @DHBA ;ADDRESS OF TRANSMIT DATA
002004 012777 000004 014256 MOV #4, @DHBAR ;START TRANSMITTER
002012 105777 014240 2$: TSTB @DHSCR ;WAIT FOR CHARACTER
002016 100375 BPL 2$ ;TO BE RECEIVED
002020 017704 014234 MOV @DHNRC, R4 ;GET RECEIVED CHARACTER
002024 020467 014326 CMP R4, TDATA ;COMPARE EXPECTED AND
002030 001401 BEQ 3$ ;RECEIVED DATA
002032 HLT 0 ;DATA ERROR
002032 104000 EMT 0
002034 104410 3$: SCOPE1 ;CHECK FOR LOOP WITH CURRENT DATA
002036 105267 014314 INCB TDATA ;UPDATE TRANSMIT DATA
002042 001352 BNE 1$
002044 104400 4$: SCOPE ;CHECK FOR ITERATIONS, LOOP
000003 LINE=LINE+1
000010 BITX=BITX+BITX
SDATA1 \LINE, \BITX

;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 3.
;CHARACTER LENGTH IS 8 BITS.
;LINE SPEED IS 9600 BAUD.

002046 TS \XN, 10, 4$, 1$
002046 012767 000340 175722 T4: MOV #340, PS ;DISABLE ALL INTERRUPTS
002054 012767 000010 014242 MOV #10, ICOUNT ;SET UP FOR 10 ITERATIONS
002062 012767 002206 014230 MOV #4$, ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <1$>
002070 012767 002132 014224 MOV #1$, FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
;ENDC
XN=XN+1
002076 012777 004000 014152 MOV @BIT11, @DHSCR ;MASTER CLEAR INTERFACE
002104 012703 000003 MOV #3, R3 ;SET UP LINE NUMBER
002110 012767 101400 014240 MOV #3*400+100000, TDATA
;SET EXPECTED LINE NUMBER
;AND VALID DATA FLAG
;EXPECTED DATA
002116 012777 000003 014132 MOV #3, @DHSCR ;SELECT LINE 3
002124 012777 033503 014130 MOV #33503, @DHLPR ;SELECT 8 BITS CHARACTER
;LENGTH, 9600 BAUD SPEED
;FOR LINE 3
002132 012777 177777 014126 1$: MOV # -1, @DHBC ;TRANSMIT 1 CHARACTER
002140 012777 016356 014116 MOV @TDATA, @DHBA ;ADDRESS OF TRANSMIT DATA
002146 012777 000010 014114 MOV #10, @DHBAR ;START TRANSMITTER
002154 105777 014076 2$: TSTB @DHSCR ;WAIT FOR CHARACTER
002160 100375 BPL 2$ ;TO BE RECEIVED
002162 017704 014072 MOV @DHNRC, R4 ;GET RECEIVED CHARACTER
002166 020467 014164 CMP R4, TDATA ;COMPARE EXPECTED AND
002172 001401 BEQ 3$ ;RECEIVED DATA
002174 HLT 0 ;DATA ERROR
002174 104000 EMT 0
002176 104410 3$: SCOPE1 ;CHECK FOR LOOP WITH CURRENT DATA
002200 105267 014152 INCB TDATA ;UPDATE TRANSMIT DATA
002204 001352 BNE 1$
002206 104400 4$: SCOPE ;CHECK FOR ITERATIONS, LOOP
000004 LINE=LINE+1

```

```

002210 000020          BITX-BITX-BITX
                                SDATA1 \LINE,\BITX

                                ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 4.
                                ;CHARACTER LENGTH IS 8 BITS.
                                ;LINE SPEED IS 9600 BAUD.

002210          TS \XN,10,4$,1$
002210 012767 000340 175560 T5:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
002216 012767 000010 014100      MOV  #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
002224 012767 002350 014066      MOV  #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB <1$>
002232 012767 002274 014062      MOV  #1$,FREEZ1     ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1

002240 012777 004000 014010      MOV  #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
002246 012703 000004 014010      MOV  #4,R3          ;SET UP LINE NUMBER
002252 012767 102000 014076      MOV  #4*400+100000,TDATA ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;SELECT LINE 4
002260 012777 000004 013770      MOV  #4,@DHSCR     ;SELECT 8 BITS CHARATER
002266 012777 033503 013766      MOV  #33503,@DHLPR ;LENGTH, 9600 BAUD SPEED
                                ;FOR LINE 4
002274 012777 177777 013764 1$:  MOV  #-1,@DHBC     ;TRANSMIT 1 CHARACTER
002302 012777 016356 013754      MOV  #TDATA,@DHBA  ;ADDRESS OF TRANSMIT DATA
002310 012777 000020 013752      MOV  #20,@DHBAR    ;START TRANSMITTER
002316 105777 013734 2$:  TSTB  @DHSCR     ;WAIT FOR CHARACTER
002322 100375          BPL  2$          ;TO BE RECEIVED
002324 017704 013730      MOV  @DHNRC,R4     ;GET RECEIVED CHARACTER
002330 020467 014022      CMP  R4,TDATA     ;COMPARE EXPECTED AND
002334 001401          BEQ  3$          ;RECEIVED DATA
002336          HLT  0          ;DATA ERROR
002336 104000          EMT  0
002340 104410          3$:  SCOPE1 ;CHECK FOR LOOP WITH CURRENT DATA
002342 105267 014010      INCB  TDATA       ;UPDATE TRANSMIT DATA
002346 001352          BNE  1$
002350 104400          4$:  SCOPE ;CHECK FOR ITERATIONS, LOOP
                                000005
                                000040
002352          BITX-BITX-BITX
                                SDATA1 \LINE,\BITX

                                ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 5.
                                ;CHARACTER LENGTH IS 8 BITS.
                                ;LINE SPEED IS 9600 BAUD.

002352          TS \XN,10,4$,1$
002352 012767 000340 175416 T6:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
002360 012767 000010 013736      MOV  #10,ICOUNT      ;SET UP FOR 10 ITERATIONS
002366 012767 002512 013724      MOV  #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

                                .IF NB <1$>
002374 012767 002436 013720      MOV  #1$,FREEZ1     ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1

002402 012777 004000 013646      MOV  #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
002410 012703 000005 013646      MOV  #5,R3          ;SET UP LINE NUMBER

```

```

002414 012767 102400 013734      MOV      #5*400+100000,TDATA
;SET EXPECTED LINE NUMBER
;AND VALID DATA FLAG
;EXPECTED DATA
002422 012777 000005 013626      MOV      #5,@DHSCR
;SELECT LINE 5
002430 012777 033503 013624      MOV      #33503,@DHLPR
;SELECT 8 BITS CHARATER
;LENGTH, 9600 BAUD SPEED
;FOR LINE 5
002436 012777 177777 013622 1$:  MOV      #-1,@DHBC
;TRANSMIT 1 CHARACTER
002444 012777 016356 013612      MOV      #TDATA,@DHBA
;ADDRESS OF TRANSMIT DATA
002452 012777 000040 013610      MOV      #40,@DHBAR
;START TRANSMITTER
002460 105777 013572 2$:  TSTB     @DHSCR
;WAIT FOR CHARACTER
002464 100375      BPL      2$
;TO BE RECEIVED
002466 017704 013566      MOV      @DHNRC,R4
;GET RECEIVED CHARACTER
002472 020467 013660      CMP      R4,TDATA
;COMPARE EXPECTED AND
002476 001401      BEQ      3$
;RECEIVED DATA
002500      HLT      0
;DATA ERROR
002500 104000      EMT      0
002502 104410      SCOPE1
;CHECK FOR LOOP WITH CURRENT DATA
002504 105267 013646      INCB     TDATA
;UPDATE TRANSMIT DATA
002510 001352      BNE     1$
002512 104400      SCOPE
;CHECK FOR ITERATIONS, LOOP
000006      LINE=LINE+1
000100      BITX=BITX+BITX
SDATA1 \LINE,\BITX

;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 6.
;CHARACTER LENGTH IS 8 BITS.
;LINE SPEED IS 9600 BAUD.

002514      TS \XN,10,4$,1$
002514 012767 000340 175254 T7:  MOV      #340,PS
;DISABLE ALL INTERRUPTS
002522 012767 000010 013574      MOV      #10,ICOUNT
;SET UP FOR 10 ITERATIONS
002530 012767 002654 013562      MOV      #4$,ESCAPE
;SET UP TO ESCAPE TO NEXT TEST
;IF NB <1$>
002536 012767 002600 013556      MOV      #1$,FREEZ1
;SET UP TO LOOP WITH DATA : 3
.ENDC
XN=XN-1
002544 012777 004000 013504      MOV      @BIT11,@DHSCR
;MASTER CLEAR INTERFACE
002552 012703 000006      MOV      #6,R3
;SET UP LINE NUMBER
002556 012767 103000 013572      MOV      #6*400+100000,TDATA
;SET EXPECTED LINE NUMBER
;AND VALID DATA FLAG
;EXPECTED DATA
002564 012777 000006 013464      MOV      #6,@DHSCR
;SELECT LINE 6
002572 012777 033503 013462      MOV      #33503,@DHLPR
;SELECT 8 BITS CHARATER
;LENGTH, 9600 BAUD SPEED
;FOR LINE 6
002600 012777 177777 013460 1$:  MOV      #-1,@DHBC
;TRANSMIT 1 CHARACTER
002606 012777 016356 013450      MOV      #TDATA,@DHBA
;ADDRESS OF TRANSMIT DATA
002614 012777 000100 013446      MOV      #100,@DHBAR
;START TRANSMITTER
002622 105777 013430 2$:  TSTB     @DHSCR
;WAIT FOR CHARACTER
002626 100375      BPL      2$
;TO BE RECEIVED
002630 017704 013424      MOV      @DHNRC,R4
;GET RECEIVED CHARACTER
002634 020467 013516      CMP      R4,TDATA
;COMPARE EXPECTED AND
002640 001401      BEQ      3$
;RECEIVED DATA
002642      HLT      0
;DATA ERROR

```

```

002642 104000          EMT      0
002644 104410          3$:     SCOPE1          ;CHECK FOR LOOP WITH CURRENT DATA
002646 105267 013504  INCB     TDATA          ;UPDATE TRANSMIT DATA
002652 001352          BNE     1$
002654 104400          4$:     SCOPE          ;CHECK FOR ITERATIONS, LOOP
          000007  LINE=LINE+1
          000200  BITX=BITX+BITX
002656          SDATA1 \LINE,\BITX

          ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 7.
          ;CHARACTER LENGTH IS 8 BITS.
          ;LINE SPEED IS 9600 BAUD.

002656          TS \XN,10,4$,1$
002656 012767 000340 175112 T10:   MOV     #340,PS          ;DISABLE ALL INTERRUPTS
002664 012767 000010 013432      MOV     #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
002672 012767 003016 013420      MOV     #4$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <1$>
002700 012767 002742 013414      MOV     #1$,FREEZ1        ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1
002706 012777 004000 013342      MOV     #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
002714 012703 000007          MOV     #7,R3              ;SET UP LINE NUMBER
002720 012767 103400 013430      MOV     #7*400+100000,TDATA
          ;SET EXPECTED LINE NUMBER
          ;AND VALID DATA FLAG
          ;EXPECTED DATA
          ;SELECT LINE 7
          ;SELECT 8 BITS CHARACTER
          ;LENGTH, 9600 BAUD SPEED
          ;FOR LINE 7
002726 012777 000007 013322      MOV     #7,@DHSCR          ;TRANSMIT 1 CHARACTER
002734 012777 033503 013320      MOV     #33503,@DHLPR     ;ADDRESS OF TRANSMIT DATA
          ;START TRANSMITTER
          ;WAIT FOR CHARACTER
          ;TO BE RECEIVED
          ;GET RECEIVED CHARACTER
          ;COMPARE EXPECTED AND
          ;RECEIVED DATA
          ;DATA ERROR
002742 012777 177777 013316 1$:     MOV     #-1,@DHBC          ;TRANSMIT 1 CHARACTER
002750 012777 016356 013306      MOV     #TDATA,@DHBA      ;ADDRESS OF TRANSMIT DATA
002756 012777 000200 013304      MOV     #200,@DHBAR       ;START TRANSMITTER
002764 105777 013266          2$:     TSTB     @DHSCR      ;WAIT FOR CHARACTER
002770 100375          BPL     2$                ;TO BE RECEIVED
002772 017704 013262      MOV     @DHNR,R4          ;GET RECEIVED CHARACTER
002776 020467 013354      CMP     R4,TDATA          ;COMPARE EXPECTED AND
003002 001401          BEQ     3$                ;RECEIVED DATA
003004          HLT     0              ;DATA ERROR
003004 104000          EMT     0
003006 104410          3$:     SCOPE1          ;CHECK FOR LOOP WITH CURRENT DATA
003010 105267 013342  INCB     TDATA          ;UPDATE TRANSMIT DATA
003014 001352          BNE     1$
003016 104400          4$:     SCOPE          ;CHECK FOR ITERATIONS, LOOP
          000010  LINE=LINE+1
          000400  BITX=BITX+BITX
003020          SDATA1 \LINE,\BITX

          ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 10.
          ;CHARACTER LENGTH IS 8 BITS.
          ;LINE SPEED IS 9600 BAUD.

003020          TS \XN,10,4$,1$
003020 012767 000340 174750 T11:   MOV     #340,PS          ;DISABLE ALL INTERRUPTS
003026 012767 000010 013270      MOV     #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
003034 012767 003160 013256      MOV     #4$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST

```

```

003042 012767 003104 013252 .IF NB <1$>
                                MOV     #1$,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
003050 012777 004000 013200 MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
003056 012703 000010 013200 MOV     #10,R3                ;SET UP LINE NUMBER
003062 012767 104000 013266 MOV     #10*400+100000,TDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;SELECT LINE 10
003070 012777 000010 013160 MOV     #10,@DHSCR           ;SELECT 8 BITS CHARATER
003076 012777 033503 013156 MOV     #33503,@DHLPR        ;LENGTH, 9600 BAUD SPEED
                                ;FOR LINE 10
                                ;TRANSMIT 1 CHARACTER
003104 012777 177777 013154 1$: MOV     #-1,@DHBC           ;ADDRESS OF TRANSMIT DATA
003112 012777 016356 013144 MOV     #TDATA,@DHBA
003120 012777 000400 013142 MOV     #400,@DHBAR        ;START TRANSMITTER
003126 105777 013124 013124 2$: TSTB   @DHSCR           ;WAIT FOR CHARACTER
003132 100375 013124 013124 BPL     2$                ;TO BE RECEIVED
003134 017704 013120 013120 MOV     @DHNRC,R4          ;GET RECEIVED CHARACTER
003140 020467 013212 013212 CMP     R4,TDATA           ;COMPARE EXPECTED AND
003144 001401 013212 013212 BEQ     3$                ;RECEIVED DATA
003146 001401 013212 013212 HLT     0                    ;DATA ERROR
003146 104000 013212 013212 EMT     0
003150 104410 013212 013212 3$: SCOPE1
003152 105267 013200 013200 INCB   TDATA              ;CHECK FOR LOOP WITH CURRENT DATA
003156 001352 013200 013200 BNE     1$                ;UPDATE TRANSMIT DATA
003160 104400 013200 013200 4$: SCOPE
                                ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
                                BITX=BITX+BITX
003162 001000 013200 013200 SDA'A1 \LINE,\BITX

                                ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 11.
                                ;CHARACTER LENGTH IS 8 BITS.
                                ;LINE SPEED IS 9600 BAUD.

003162 012767 000340 174606 TS \XN,10,4$,1$
003170 012767 000010 013126 T12: MOV     #340,PS        ;DISABLE ALL INTERRUPTS
003176 012767 003322 013114 MOV     #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
                                MOV     #4$,ESCAPE          ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
003204 012767 003246 013110 MOV     #1$,FREEZ1           ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
003212 012777 004000 013036 MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
003220 012703 000011 013036 MOV     #11,R3                ;SET UP LINE NUMBER
003224 012767 104400 013124 MOV     #11*400+100000,TDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;SELECT LINE 11
003232 012777 000011 013016 MOV     #11,@DHSCR           ;SELECT 8 BITS CHARATER
003240 012777 033503 013014 MOV     #33503,@DHLPR        ;LENGTH, 9600 BAUD SPEED
                                ;FOR LINE 11
                                ;TRANSMIT 1 CHARACTER
003246 012777 177777 013012 1$: MOV     #-1,@DHBC           ;ADDRESS OF TRANSMIT DATA
003254 012777 016356 013002 MOV     #TDATA,@DHBA
003262 012777 001000 013000 MOV     #1000,@DHBAR        ;START TRANSMITTER

```

```

003270 105777 012762      2$:  TSTB  @DHSCR          ;WAIT FOR CHARACTER
003274 100375              BPL  2$                ;TO BE RECEIVED
003276 017704 012756      MOV  @DHNRC,R4         ;GET RECEIVED CHARACTER
003302 020467 013050      CMP  R4,TDATA         ;COMPARE EXPECTED AND
003306 001401              BEQ  3$                ;RECEIVED DATA
003310              HLT  0                ;DATA ERROR
003310 104000              EMT  0
003312 104410      3$:  SCOPE1          ;CHECK FOR LOOP WITH CURRENT DATA
003314 105267 013036      INCB TDATA            ;UPDATE TRANSMIT DATA
003320 001352              BNE  1$
003322 104400      4$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
      000012      LINE=LINE+1
      002000      BITX=BITX+BITX
003324      SDATA1 \LINE,\BITX

      ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 12.
      ;CHARACTER LENGTH IS 8 BITS.
      ;LINE SPEED IS 9600 BAUD.

003324 012767 000340 174444  TS \XN,10,4$,1$
003324 012767 000010 012764  T13:  MOV  #340,PS          ;DISABLE ALL INTERRUPTS
003332 012767 000010 012764      MOV  #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
003340 012767 003464 012752      MOV  #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <1$>
003346 012767 003410 012746      MOV  #1$,FREEZ1      .SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
003354 012777 004000 012674      MOV  #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
003362 012703 000012              MOV  #12,R3          ;SET UP LINE NUMBER
003366 012767 105000 012762      MOV  #12*400+100000,TDATA
      ;SET EXPECTED LINE NUMBER
      ;AND VALID DATA FLAG
      ;EXPECTED DATA
      ;SELECT LINE 12
      ;SELECT 8 BITS CHARACTER
      ;LENGTH, 9600 BAUD SPEED
      ;FOR LINE 12
003374 012777 000012 012654      MOV  #12,@DHSCR     ;TRANSMIT 1 CHARACTER
003402 012777 033503 012652      MOV  #33503,@DHLPR ;ADDRESS OF TRANSMIT DATA
      ;START TRANSMITTER
003410 012777 177777 012650  1$:  MOV  #-1,@DHBC     ;WAIT FOR CHARACTER
003416 012777 016356 012640      MOV  #TDATA,@DHBA  ;TO BE RECEIVED
003424 012777 002000 012636      MOV  #2000,@DHBAR  ;GET RECEIVED CHARACTER
003432 105777 012620      2$:  TSTB  @DHSCR     ;COMPARE EXPECTED AND
003436 100375              BPL  2$                ;RECEIVED DATA
003440 017704 012614      MOV  @DHNRC,R4     ;DATA ERROR
003444 020467 012706      CMP  R4,TDATA
003450 001401              BEQ  3$
003452              HLT  0
003452 104000              EMT  0
003454 104410      3$:  SCOPE1          ;CHECK FOR LOOP WITH CURRENT DATA
003456 105267 012674      INCB TDATA            ;UPDATE TRANSMIT DATA
003462 001352              BNE  1$
003464 104400      4$:  SCOPE          ;CHECK FOR ITERATIONS, LOOP
      000013      LINE=LINE+1
      004000      BITX=BITX+BITX
003466      SDATA1 \LINE,\BITX

      ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 13.
      ;CHARACTER LENGTH IS 8 BITS.

```


;LINE SPEED IS 9600 BAUD.

```

003466      TS \XN,10,4$,1$
003466 012767 000340 174302 T14:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
003474 012767 000010 012622      MOV    #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
003502 012767 003626 012610      MOV    #4$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <1$>
003510 012767 003552 012604      MOV    #1$,FREEZ1       ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
003516 012777 004000 012532      MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
003524 012703 000013 012620      MOV    #13,R3           ;SET UP LINE NUMBER
003530 012767 105400 012620      MOV    #13*400+100000,TDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
003536 012777 000013 012512      MOV    #13,@DHSCR      ;SELECT LINE 13
003544 012777 033503 012510      MOV    #33503,@DHLPR   ;SELECT 8 BITS CHARACTER
                                ;LENGTH, 9600 BAUD SPEED
                                ;FOR LINE 13
003552 012777 177777 012506 1$:  MOV    #-1,@DHBC       ;TRANSMIT 1 CHARACTER
003560 012777 016356 012476      MOV    #TDATA,@DH9A    ;ADDRESS OF TRANSMIT DATA
003566 012777 004000 012474      MOV    #4000,@DH8AR    ;START TRANSMITTER
003574 105777 012456 012474 2$:  TSTB  @DHSCR          ;WAIT FOR CHARACTER
003600 100375 012456 012474      BPL    2$              ;TO BE RECEIVED
003602 017704 012452 012474      MOV    @DHNR,R4        ;GET RECEIVED CHARACTER
003606 020467 012544 012474      CMP    R4,TDATA        ;COMPARE EXPECTED AND
003612 001401 012544 012474      BEQ    3$              ;RECEIVED DATA
003614 010400 012544 012474      HLT    0               ;DATA ERROR
003614 104410 012544 012474      EMT    0
003616 104410 012544 012474 3$:  SCOPE1
003620 105267 012532 012474      INCB  TDATA            ;CHECK FOR LOOP WITH CURRENT DATA
003624 001352 012532 012474      BNE   1$              ;UPDATE TRANSMIT DATA
003626 10~400 012532 012474 4$:  SCOPE
                                ;CHECK FOR ITERATIONS. LOOP
                                LINE=LINE+1
                                BITX=BITX+BITX
003630      SDATA1 \LINE,\BITX

```

```

;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 14.
;CHARACTER LENGTH IS 8 BITS.
;LINE SPEED IS 9600 BAUD.

```

```

003630      TS \XN,10,4$,1$
003630 012767 000340 174140 T15:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
003636 012767 000010 012460      MOV    #10,ICOUNT        ;SET UP FOR 10 ITERATIONS
003644 012767 003770 012446      MOV    #4$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB  <1$>
003652 012767 003714 012442      MOV    #1$,FREEZ1       ;SET UP TO LOOP WITH DATA           ; 3
                                .ENDC
                                XN=XN+1
003660 012777 004000 012370      MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
003666 012703 000014 012456      MOV    #14,R3           ;SET UP LINE NUMBER
003672 012767 106000 012456      MOV    #14*400+100000,TDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
003700 012777 000014 012350      MOV    #14,@DHSCR      ;SELECT LINE 14

```

```

003706 012777 033503 012346      MOV      #33503, @DHLPR      ;SELECT 8 BITS CHARATER
                                           ;LENGTH, 9600 BAUD SPEED
                                           ;FOR LINE 14
003714 012777 177777 012344 1$:  MOV      #-1, @DHBC      ;TRANSMIT 1 CHARACTER
003722 012777 016356 012334      MOV      #TDATA, @DHBA     ;ADDRESS OF TRANSMIT DATA
003730 012777 010000 012332      MOV      #10000, @DHBAR    ;START TRANSMITTER
003736 105777 012314      2$:  TSTB      @DHSCR      ;WAIT FOR CHARACTER
003742 100375      BPL      2$              ;TO BE RECEIVED
003744 017704 012310      MOV      @DHNRC, R4       ;GET RECEIVED CHARACTER
003750 020467 012402      CMP      R4, TDATA       ;COMPARE EXPECTED AND
003754 001401      BEQ      3$              ;RECEIVED DATA
003756      HLT      0              ;DATA ERROR
                                           EMT      0
003760 104410      3$:  SCOPE1      ;CHECK FOR LOOP WITH CURRENT DATA
003762 105267 012370      INCB     TDATA          ;UPDATE TRANSMIT DATA
003766 001352      BNE     1$
003770 104400      4$:  SCOPE      ;CHECK FOR ITERATIONS, LOOP
      000015      LINE=LINE+1
      020000      BITX=BITX+BITX
      SDATA1 \LINE, \BITX

                                           ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 15.
                                           ;CHARACTER LENGTH IS 8 BITS.
                                           ;LINE SPEED IS 9600 BAUD.

003772      TS \XN, 10, 4$, 1$
003772 012767 000340 173776 T16:  MOV      #340, PS      ;DISABLE ALL INTERRUPTS
004000 012767 000010 012316      MOV      #10, ICOUNT     ;SET UP FOR 10 ITERATIONS
004006 012767 004132 012304      MOV      #4$, ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <1$>
004014 012767 004056 012300      MOV      #1$, FREEZ1     ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
004022 012777 004000 012226      MOV      @BIT11, @DHSCR  ;MASTER CLEAR INTERFACE
004030 012703 000015      MOV      #15, R3        ;SET UP LINE NUMBER
004034 012767 106400 012314      MOV      #15*400+100000, TDATA
                                           ;SET EXPECTED LINE NUMBER
                                           ;AND VALID DATA FLAG
                                           ;EXPECTED DATA
004042 012777 000015 012206      MOV      #15, @DHSCR    ;SELECT LINE 15
004050 012777 033503 012204      MOV      #33503, @DHLPR ;SELECT 8 BITS CHARATER
                                           ;LENGTH, 9600 BAUD SPEED
                                           ;FOR LINE 15
004056 012777 177777 012202 1$:  MOV      #-1, @DHBC      ;TRANSMIT 1 CHARACTER
004064 012777 016356 012172      MOV      #TDATA, @DHBA   ;ADDRESS OF TRANSMIT DATA
004072 012777 020000 012170      MOV      #20000, @DHBAR  ;START TRANSMITTER
004100 105777 012152      2$:  TSTB      @DHSCR      ;WAIT FOR CHARACTER
004104 100375      BPL      2$              ;TO BE RECEIVED
004106 017704 012146      MOV      @DHNRC, R4     ;GET RECEIVED CHARACTER
004112 020467 012240      CMP      R4, TDATA     ;COMPARE EXPECTED AND
004116 001401      BEQ      3$              ;RECEIVED DATA
004120      HLT      0              ;DATA ERROR
                                           EMT      0
004122 104410      3$:  SCOPE1      ;CHECK FOR LOOP WITH CURRENT DATA
004124 105267 012226      INCB     TDATA          ;UPDATE TRANSMIT DATA
004130 001352      BNE     1$
004132 104400      4$:  SCOPE      ;CHECK FOR ITERATIONS, LOOP

```

```

000016
000000
004134 LINE=LINE+1
        BITX=BITX+BITX
        SDATA1 \LINE,\BITX

        ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 16.
        ;CHARACTER LENGTH IS 8 BITS.
        ;LINE SPEED IS 9600 BAUD.

004134 TS \XN,10,4$,1$
004134 012767 000340 173634 T17: MOV #340,PS ;DISABLE ALL INTERRUPTS
004142 012767 000010 012154 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
004150 012767 004274 012142 MOV #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

        .IF NB <1$>
004156 012767 004220 012136 MOV #1$,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
        .ENDC
        XN=XN+1

004164 012777 004000 012064 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
004172 012703 000016 MOV #16,R3 ;SET UP LINE NUMBER
004176 012767 107000 012152 MOV #16*400+100000,TDATA ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;SELECT LINE 16
                                ;SELECT 8 BITS CHARACTER
                                ;LENGTH, 9600 BAUD SPEED
                                ;FOR LINE 16
                                ;TRANSMIT 1 CHARACTER
                                ;ADDRESS OF TRANSMIT DATA
                                ;START TRANSMITTER
                                ;WAIT FOR CHARACTER
                                ;TO BE RECEIVED
                                ;GET RECEIVED CHARACTER
                                ;COMPARE EXPECTED AND
                                ;RECEIVED DATA
                                ;DATA ERROR

004204 012777 000016 012044 MOV #16,@DHSCR
004212 012777 033503 012042 MOV #33503,@DHLPR

004220 012777 177777 012040 1$: MOV #-1,@DHBC ;TRANSMIT 1 CHARACTER
004226 012777 016356 012030 MOV #TDATA,@DHBA ;ADDRESS OF TRANSMIT DATA
004234 012777 040000 012026 MOV #40000,@DHBAR ;START TRANSMITTER
004242 105777 012010 2$: TSTB @DHSCR ;WAIT FOR CHARACTER
004246 100375 BPL 2$ ;TO BE RECEIVED
004250 017704 012004 MOV @DHNRC,R4 ;GET RECEIVED CHARACTER
004254 020467 012076 CMP R4,TDATA ;COMPARE EXPECTED AND
004260 001401 BEQ 3$ ;RECEIVED DATA
004262 HLT 0 ;DATA ERROR
004262 104000 EMT 0
004264 104410 3$: SCOPE1 ;CHECK FOR LOOP WITH CURRENT DATA
004266 105267 012064 INCB TDATA ;UPDATE TRANSMIT DATA
004272 001352 BNE 1$
004274 104400 4$: SCOPE ;CHECK FOR ITERATIONS, LOOP
        000017
        100000
004276 SDATA1 \LINE,\BITX

        ;TRANSMIT ALL CHARACTERS ONE AT A TIME ON LINE 17.
        ;CHARACTER LENGTH IS 8 BITS.
        ;LINE SPEED IS 9600 BAUD.

004276 TS \XN,10,4$,1$
004276 012767 000340 173472 T20: MOV #340,PS ;DISABLE ALL INTERRUPTS
004304 012767 000010 012012 MOV #10,ICOUNT ;SET UP FOR 10 ITERATIONS
004312 012767 004436 012000 MOV #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST

        .IF NB <1$>
004320 012767 004362 011774 MOV #1$,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
        .ENDC
        XN=XN+1

004326 012777 004000 011722 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE

```

```

004334 012703 000017          MOV    #17,R3          ;SET UP LINE NUMBER
004340 012767 107400 012010  MOV    #17*400+100000,TDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
004346 012777 000017 011702  MOV    #17,@DHSCR     ;SELECT LINE 17
004354 012777 033503 011700  MOV    #33503,@DHLPR  ;SELECT 8 BITS CHARATER
                                ;LENGTH, 9600 BAUD SPEED
                                ;FOR LINE 17
004362 012777 171777 011676 1$:  MOV    #-1,@DHBC      ;TRANSMIT 1 CHARACTER
004370 012777 016356 011666  MOV    #TDATA,@DHBA   ;ADDRESS OF TRANSMIT DATA
004376 012777 100000 011664  MOV    #100000,@DHBAR ;START TRANSMITTER
004404 105777 011646          2$:  TSTB   @DHSCR       ;WAIT FOR CHARACTER
004410 100375          BPL    2$             ;TO BE RECEIVED
004412 017704 011642          MOV    @DHNRC,R4      ;GET RECEIVED CHARACTER
004416 020467 011734          CMP    R4,TDATA      ;COMPARE EXPECTED AND
004422 001401          BEQ    3$             ;RECEIVED DATA
004424          HLT    0              ;DATA ERROR
004424 104000          EMT    0
004426 104410          3$:  SCOPE1
004430 105267 011722          INCB   TDATA
004434 001352          BNE    1$
004436 104400          4$:  SCOPE
                                ;CHECK FOR LOOP WITH CURRENT DATA
                                ;UPDATE TRANSMIT DATA
                                ;CHECK FOR ITERATIONS, LOOP
                                LINE=LINE+1
                                BITX=BITX+BITX
                                LINE=0
                                XLINE=LINE
                                BITX=1
                                XBIT=BITX
                                .REPT 20
                                SDATA2 \LINE,\BITX
                                .NLIST
                                LINE=LINE+1
                                BITX=BITX+BITX
                                .LIST
                                .ENDR
                                SDATA2 \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 0
                                ;CHARATER LENGTH IS 8 BITS
                                ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
                                ;TO 9600 BAUD.
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH SPEED
17      000000
18      000000
19      000001
20      000001
22      000020
23
24
25
26
27
28
004440          TS \XN,1,4$,1$
004440 012767 000340 173330 T21:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
004446 012767 000001 011650  MOV    #1,ICOUNT       ;SET UP FOR 1 ITERATIONS
004454 012767 004634 011636  MOV    #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
004462 012767 004524 011632  MOV    #1$,FREEZ1     ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
004470 012777 004000 011560  MOV    #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
004476 012702 000001          MOV    #1,R2           ;FIRST SPEED CODE
004502 012705 000000          MOV    #0,R5           ;LINE 0 WILL BE TESTED

```

```

004506 012767 100000 011644      MOV      #0*400+100000,RDATA      ;SET EXPECTED LINE NUMBER
                                           ;AND VALID DATA FLAG
                                           ;EXPECTED DATA
004514 012700 000015      MOV      #15,R0                  ;13 SPEEDS WILL BE TESTED
004520 012701 002103      MOV      #2103,R1                ;FIRST SPEED =50 BAUD,
                                           ;8 BITS PER CHARACTER
004524 010577 011526      1$:   MOV      R5,@DHSCR          ;SELECT LINE 0
004530 010177 011526      MOV      R1,@DHLPR              ;SET LINE SPEED AND
                                           ;CHARACTER LENGTH
004534 012777 016364 011522      MOV      #TBUF,@DHBA            ;ADDRESS OF TRANSMITTER
                                           ;DATA BUFFER
004542 012777 177400 011516      MOV      #-400,@DHBC            ;400 (OCTAL) BYTES
                                           ;WILL BE TRANSMITTED
004550 012777 000001 011512      MOV      #1,@DHBAR              ;START TRANSMITTER
004556 105777 011474      2$:   TSTB      @DHSCR            ;WAIT FOR DATA TO BE RECEIVED
004562 100375      BPL      2$
004564 017703 011470      MOV      @DHNRC,R3              ;GET RECEIVED DATA
004570 020367 011564      CMP      R3,RDATA               ;COMPER EXPECTED AND RECEIVED DATA
004574 001407      BEQ      3$
004576 005077 011466      CLR      @DHBAR                  ;STOP TRANSMITTER
004602      HLT      1                      ;DATA ERROR
004602 104001      EMT      1
004604 104410      SCOPE1
004606 012777 000001 011454      MOV      #1,@DHBAR              ;CHECK FOR LOOP AT CURRENT SPEED
004614 105267 011540      3$:   INCB      RDATA              ;RESTART TRANSMITTER
                                           ;UPDATA EXPECTED DATA
004620 001356      BNE      2$
004622 062701 002100      ADD      #2100,R1                ;UPDATE LINE SPEED
004626 005202      INC      R2                      ;UPDATE SPEED CODE
004630 005300      DEC      R0
004632 001334      BNE      1$
004634 104400      4$:   SCOPE
000001      LINE=LINE+1
000002      BITX=BITX+BITX
004636      SDATA2 \LINE,\BITX

                                           ;SINGLE LINE DATA TEST
                                           ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 1
                                           ;CHARATER LENGTH IS 8 BITS
                                           ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
                                           ;TO 9600 BAUD.
                                           ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                           ;AT EACH SPEED

004636      TS \XN,1,4$,1$
004636 012767 000340 173132      T22:  MOV      #340,PS            ;DISABLE ALL INTERRUPTS
004644 012767 000001 011452      MOV      #1,ICOUNT              ;SET UP FOR 1 ITERATIONS
004652 012767 005032 011440      MOV      #4$,ESCAPE             ;SET UP TO ESCAPE TO NEXT TEST
                                           ;IF NB <1$>
004660 012767 004722 011434      MOV      #1$,FREEZ1             ;SET UP TO LOOP WITH DATA      ; 3
                                           ;.ENDC
                                           XN=XN+1
004666 012777 004000 011362      MOV      @BIT11,@DHSCR          ;MASTER CLEAR INTERFACE
004674 012702 000001      MOV      #1,R2                  ;FIRST SPEED CODE
004700 012705 000001      MOV      #1,R5                  ;LINE 1 WILL BE TESTED
004704 012767 1r0400 011446      MOV      #1*400+100000,RDATA    ;SET EXPECTED LINE NUMBER

```

```

004712 012700 000015      MOV    #15,R0      ;AND VALID DATA FLAG
004716 012701 002103      MOV    #2103,R1    ;EXPECTED DATA
                                ;13 SPEEDS WILL BE TESTED
004722 010577 011330      1$:  MOV    R5,@DHSCR ;FIRST SPEED =50 BAUD,
004726 010177 011330      MOV    R1,@DHLPR   ;8 BITS PER CHARACTER
                                ;SELECT LINE 1
004732 012777 016364 011324  MOV    #TBUF,@DHBA ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
004740 012777 177400 011320  MOV    #-400,@DHBC ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
004746 012777 000002 011314  MOV    #2,@DHBAR   ;START TRANSMITTER
004754 105777 011276      2$:  TSTB   @DHSCR     ;WAIT FOR DATA TO BE RECEIVED
004760 100375                BPL    2$
004762 017703 011272      MOV    @DHNRC,R3   ;GET RECEIVED DATA
004766 020367 011366      CMP    R3,RDATA    ;COMPER EXPECTED AND RECEIVED DATA
004772 001407                BEQ    3$
004774 005077 011270      CLR    @DHBAR
005000                HLT    1
005000                EMT    1
                                ;STOP TRANSMITTER
                                ;DATA ERROR
005002 104410                SCOPE1 ;CHECK FOR LOOP AT CURRENT SPEED
005004 012777 000002 011256  MOV    #2,@DHBAR   ;RESTART TRANSMITTER
005012 105267 011342      3$:  INCB   RDATA      ;UPDATA EXPECTED DATA
005016 001356                BNE    2$
005020 062701 002100      ADD    #2100,R1    ;UPDATE LINE SPEED
005024 005202                INC    R2           ;UPDATE SPEED CODE
005026 005300                DEC    R0
005030 001334                BNE    1$
005032 104400                4$:  SCOPE ;
                                LINE=LINE+1
                                BITX=BITX+BITX
005034 000002                SDATA2 \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 2
                                ;CHARATER LENGTH IS 8 BITS
                                ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
                                ;TO 9600 BAUD.
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH SPEED

005034 000024                TS \XN,1,4$,1$
005034 012767 000340 172734 T23:  MOV    #340,PS   ;DISABLE ALL INTERRUPTS
005042 012767 000001 011254      MOV    #1,ICOUNT   ;SET UP FOR 1 ITERATTIONS
005050 012767 005230 011242      MOV    #4$,ESCAPE  ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
005056 012767 005120 011236      MOV    #1$,FREEZ1  ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
005064 012777 004000 011164      MOV    @BIT11,@DHSCR ;MASTER CLEAR INTERFACE
005072 012702 000001                MOV    #1,R2       ;FIRST SPEED CODE
005076 012705 000002                MOV    #2,R5       ;LINE 2 WILL BE TESTED
005102 012767 101000 011250      MOV    #2*400+100000,RDATA ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA

```

```

005110 012700 000015      MOV      #15,R0      ;13 SPEEDS WILL BE TESTED
005114 012701 002103      MOV      #2103,R1    ;FIRST SPEED =50 BAUD,
                                ;8 BITS PER CHARACTER
005120 010577 011132      1$:      MOV      R5,@DHSCR ;SELECT LINE 2
005124 010177 011132      MOV      R1,@DHLPR  ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
005130 012777 016364 011126      MOV      @TBUF,@DHBA ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
005136 012777 177400 011122      MOV      #-400,@DHBC ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
005144 012777 000004 011116      MOV      #4,@DHBAR   ;START TRANSMITTER
005152 105777 011100      2$:      TSTB     @DHSCR     ;WAIT FOR DATA TO BE RECEIVED
005156 100375                BPL      2$
005160 017703 011074      MOV      @DHNRC,R3   ;GET RECEIVED DATA
005164 020367 011170      CMP      R3,RDATA    ;COMPER EXPECTED AND RECEIVED DATA
005170 001407                BEQ      3$
005172 005077 011072      CLR      @DHBAR     ;STOP TRANSMITTER
005176                HLT      1      ;DATA ERROR
005176 104001                EMT      1
005200 104410                SCOPE1
005202 012777 000004 011060      MOV      #4,@DHBAR   ;CHECK FOR LOOP AT CURRENT SPEED
005210 105267 011144      3$:      INCB     RDATA      ;RESTART TRANSMITTER
005214 001356                BNE     2$           ;UPDATA EXPECTED DATA
005216 062701 002100      ADD     #2100,R1     ;UPDATE LINE SPEED
005222 005202                INC     R2           ;UPDATE SPEED CODE
005224 005300                DEC     R0
005226 001334                BNE     1$
005230 104400      4$:      SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
005232 000003      SDATA2 \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 3
                                ;CHARATER LENGTH IS 8 BITS
                                ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
                                ;TO 9600 BAUD.
                                ;A BLOCK OF 400 CHARACTER, WILL BE TRANSMITTED
                                ;AT EACH SPEED

005232                TS \XN,1,4$,1$
005232 012767 000340 172536      T24:     MOV      #340,PS ;DISABLE ALL INTERRUPTS
005240 012767 000001 011056      MOV      #1,ICOUNT   ;SET UP FOR 1 ITERATIONS
005246 012767 005426 011044      MOV      #4$,ESCAPE  ;SET UP TO ESCAPE TO NEXT TEST
                                .IF #8 <1$>
005254 012767 005316 011040      MOV      #1$,FREEZ1  ;SET UP TO LOOP WITH DATA ; 3
                                .ENDC
                                XN=XN+1
005262 012777 004000 010766      MOV      @BIT11,@DHSCR ;MASTER CLEAR INTERFACE
005270 012702 000001                MOV      #1,R2      ;FIRST SPEED CODE
005274 012705 000003                MOV      #3,R5      ;LINE 3 WILL BE TESTED
005300 012767 101400 011052      MOV      #3*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
005306 012700 000015      MOV      #15,R0      ;13 SPEEDS WILL BE TESTED
005312 012701 002103      MOV      #2103,R1    ;FIRST SPEED =50 BAUD.

```

```

005316 010577 010734      1$:  MOV    R5,@DHSCR      ;8 BITS PER CHARACTER
005322 010177 010734      MCV    R1,@DHLPR      ;SELECT LINE 3
                                ;SET LINE SPEED AND
005326 012777 016364 010730  MOV    #TBUF,@DHBA    ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
005334 012777 177400 010724  MOV    #-400,@DHBC    ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
005342 012777 000010 010720  MOV    #10,@DHBAR     ;START TRANSMITTER
005350 105777 010702      2$:  TSTB   @DHSCR         ;WAIT FOR DATA TO BE RECEIVED
005354 100375              BPL    2$
005356 017703 010676      MOV    @DHNRC,R3      ;GET RECEIVED DATA
005362 020367 010772      CMP    R3,RDATA      ;COMPER EXPECTED AND RECEIVED DATA
005366 001407              BEQ    3$
005370 005077 010674      CLR    @DHBAR        ;STOP TRANSMITTER
005374              HLT    1
                                ;DATA ERROR
005374 104001              EMT    1
005376 104410              SCOPE1
005400 012777 000010 010662  MOV    #10,@DHBAR     ;CHECK FOR LOOP AT CURRENT SPEED
005406 105267 010746      3$:  INCB   RDATA         ;RESTART TRANSMITTER
                                ;UPDATA EXPECTED DATA
005412 001356              BNE    2$
005414 062701 002100      ADD    #2100,R1      ;UPDATE LINE SPEED
005420 005202              INC    R2            ;UPDATE SPEED CODE
005422 005300              DEC    R0
005424 001334              BNE    1$
005426 104400      4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
005430      SDATA2 \LINE,\BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 4
                                ;CHARATER LENGTH IS 8 BITS
                                ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
                                ;TO 9600 BAUD.
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH SPEED

005430      TS \XN,1,4$,1$
005430 012767 000340 172340  T25:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
005436 012767 000001 010660      MOV    #1,ICOUNT     ;SET UP FOR 1 ITERATIONS
005444 012767 005624 010646      MOV    #4$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST

005452 012767 005514 010642      .IF NB <1$>
                                MOV    #1$,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1

005460 012777 004000 010570      MOV    @BIT11,@DHSCR ;MASTER CLEAR INTERFACE
005466 012702 000001              MOV    #1,R2         ;FIRST SPEED CODE
005472 012705 000004              MOV    #4,R5         ;LINE 4 WILL BE TESTED
005476 012767 102000 010654      MOV    #4*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
005504 012700 000015              MOV    #15,R0        ;13 SPEEDS WILL BE TESTED
005510 012701 002103              MOV    #2103,R1     ;FIRST SPEED =50 BAUD.
005514 010577 010536      1$:  MOV    R5,@DHSCR     ;8 BITS PER CHARACTER
                                ;SELECT LINE 4

```



```

005520 010177 010536          MOV      R1,@DHLPR          ;SET LINE SPEED AND
005524 012777 016364 010532    MOV      @TBUF,@DHBA        ;CHARACTER LENGTH
005532 012777 177400 010526    MOV      #-400,@DHBC        ;ADDRESS OF TRANSMITTER
005540 012777 000020 010522    MOV      @20,@DHBAR         ;DATA BUFFER
005546 105777 010504          2$:    TSTB    @DHSCR           ;400 (OCTAL) BYTES
005552 100375                    BPL      2$                 ;WILL BE TRANSMITTED
005554 017703 010500          MOV      @DHNRC,R3          ;START TRANSMITTER
005560 020367 010574          CMP      R3,RDATA          ;WAIT FOR DATA TO BE RECEIVED
005564 001407                    BEQ      3$                 ;GET RECEIVED DATA
005566 005077 010476          CLR      @DHBAR            ;COMPER EXPECTED AND RECEIVED DATA
005572                    HLT      1                   ;STOP TRANSMITTER
005572 104001                    EMT      1                   ;DATA ERROR
005574 104410                    SCOPE1
005576 012777 000020 010464    MOV      @20,@DHBAR         ;CHECK FOR LOOP AT CURRENT SPEED
005604 105267 010550          3$:    INCB    RDATA           ;RESTART TRANSMITTER
005610 001356                    BNE      2$                 ;UPDATA EXPECTED DATA
005612 062701 002100          ADD      @2100,R1           ;UPDATE LINE SPEED
005616 005202                    INC      R2                 ;UPDATE SPEED CODE
005620 005300                    DEC      R0
005622 001334                    BNE      1$
005624 104400                    4$:    SCOPE
000005          LINE=LINE+1
000040          BITX=BITX+BITX
005626          SDATA2 \LINE,\BITX

          ;SINGLE LINE DATA TEST
          ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 5
          ;CHARATER LENGTH IS 8 BITS
          ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
          ;TO 9600 BAUD.
          ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
          ;AT EACH SPEED

005626          TS \XN,1,4$,1$
005626 012767 000340 172142    T26:    MOV      @340,PS        ;DISABLE ALL INTERRUPTS
005634 012767 000001 010462    MOV      @1,ICOUNT         ;SET UP FOR 1 ITERATIONS
005642 012767 006022 010450    MOV      @4$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
005650 012767 005712 010444    .IF NB <1$>
005650          MOV      @1$,FREEZ1        ;SET UP TO LOOP WITH DATA          ; 3
005650          .ENDC
000027          XN=XN+1
005656 012777 004000 010372    MOV      @BIT11,@DHSCR     ;MASTER CLEAR INTERFACE
005664 012702 000001          MOV      @1,R2             ;FIRST SPEED CODE
005670 012705 000005          MOV      @5,R5             ;LINE 5 WILL BE TESTED
005674 012767 102400 010456    MOV      @5*400+100000,RDATA
005674          ;SET EXPECTED LINE NUMBER
005674          ;AND VALID DATA FLAG
005674          ;EXPECTED DATA
005702 012700 000015          MOV      @15,R0            ;13 SPEEDS WILL BE TESTED
005706 012701 002103          MOV      @2103,R1         ;FIRST SPEED =50 BAUD,
005712 010577 010340          1$:    MOV      R5,@DHSCR       ;8 BITS PER CHARACTER
005716 010177 010340          MOV      R1,@DHLPR        ;SELECT LINE 5
005716          ;SET LINE SPEED AND
005716          ;CHARACTER LENGTH

```

```

005722 012777 016364 010334      MOV    #TBUF, @DHBA      ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
005730 012777 177400 010330      MOV    #-400, @DHBC     ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
005736 012777 000040 010324      MOV    #40, @DHBAR      ;START TRANSMITTER
005744 105777 010306              2$:  TSTB   @DHSCR          ;WAIT FOR DATA TO BE RECEIVED
005750 100375                      BPL    2$
005752 017703 010302              MOV    @DHnRC, R3       ;GET RECEIVED CA A
005756 020367 010376              CMP    R3, RDATA       ;COMPER EXPECTED AND RECEIVED DATA
005762 001407                      BEQ    3$
005764 005077 010300              CLR    @DHBAR          ;STOP TRANSMITTEP
005770                      HLT    1                  ;DATA ERROR
005770 104001                      EMT    1
005772 104410                      SCOPE1
005774 012777 000040 010266      MOV    #40, @DHBAR      ;CHECK FOR LOOP AT CURRENT SPEED
006002 105267 010352              3$:  INCB   RDATA          ;RESTART TRANSMITTER
006006 001356                      BNE    2$              ;UPDATA EXPECTED DATA
006010 062701 002100              ADD    #2100, R1       ;UPDATE LINE SPEED
006014 005202                      INC    R2              ;UPDATE SPEED CODE
006016 005300                      DEC    R0
006020 001334                      BNE    1$
006022 104400              4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
0C6024 000006
                                SDATA2 \LINE, \BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 6
                                ;CHARATER LENGTH IS 8 BITS
                                ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
                                ;TO 9600 BAUD.
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH SPEED

006024 000030      TS \XN, 1, 4$, 1$
006024 012767 000340 171744      T27:  MOV    #340, PS      ;DISABLE ALL INTERRUPTS
006032 012767 000001 010264      MOV    #1, ICOUNT      ;SET UP FOR 1 ITERATIONS
006040 012767 006220 010252      MOV    #4$, ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
006046 012767 006110 010246      MOV    #1$, FREEZ1     ;SET UP TO LOOP WITH DATA      : 3
                                .ENDC
                                XN=XN+1
006054 012777 004000 010174      MOV    #BIT11, @DHSCR  ;MASTER CLEAR INTERFACE
006062 012702 000001              MOV    #1, R2          ;FIRST SPEED CODE
006066 012705 000006              MOV    #6, R5          ;LINE 6 WILL BE TESTED
006072 012767 103000 010260      MOV    #6*400+100000, RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
006100 012700 000015              MOV    #15, R0        ;13 SPEEDS WILL BE TESTED
006104 012701 002103              MOV    #2103, R1     ;FIRST SPEED =50 BAUD,
                                ;8 BITS PER CHARACTER
006110 010577 010142              1$:  MOV    R5, @DHSCR   ;SELECT LINE 6
006114 010177 010142              MOV    R1, @DHLPR    ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
006120 012777 016364 010136      MOV    #TBUF, @DHBA  ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER

```

```

006126 012777 177400 010132      MOV      # 400, @DHBC      ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
006134 012777 000100 010126      MOV      #100, @DHBAR     ;START TRANSMITTER
006142 105777 010110              2$:    TSTB      @DHSCR      ;WAIT FOR DATA TO BE RECEIVED
006146 100375                      BPL      2$
006150 017703 010104      MOV      @DHNR, R3      ;GET RECEIVED DATA
006154 020367 010200      CMP      R3, RDATA     ;COMPER EXPECTED AND RECEIVED DATA
006160 001407                      BEQ      3$
006162 005077 010102      CLR      @DHBAR      ;STOP TRANSMITTER
006166                      HLT      1      ;DATA ERROR
006166 104001                      EMT      1
006170 104410                      SCOPE1
006172 012777 000100 010070      MOV      #100, @DHBAR     ;CHECK FOR LOOP AT CURRENT SPEED
006200 105267 010154              3$:    INCB      RDATA     ;RESTART TRANSMITTER
006204 001356                      BNE      2$      ;UPDATA EXPECTED DATA
006206 062701 002100      ADD      #2100, R1     ;UPDATE LINE SPEED
006212 005202                      INC      R2      ;UPDATE SPEED CODE
006214 005300                      DEC      R0
006216 001334                      BNE      1$
006220 104400              4$:    SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
006222 000007      SDATA2 \LINE, \BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 7
                                ;CHARATER LENGTH IS 8 BITS
                                ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
                                ;TO 9600 BAUD.
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH SPEED

006222 012767 000340 171546      TS \XN, 1, 4$, 1$
006222 012767 000001 010066      T30:    MOV      #340, PS      ;DISABLE ALL INTERRUPTS
006230 012767 000001 010066      MOV      #1, ICOUNT     ;SET UP FOR 1 ITERATIONS
006236 012767 006416 010054      MOV      #4$, ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
006244 012767 006306 010050      MOV      #1$, FREEZ1    ;SET UP TO LOOP WITH DATA      : 3
                                .ENDC
                                XN=XN+1
006252 012777 004000 007776      MOV      #BIT11, @DHSCR ;MASTER CLEAR INTERFACE
006260 012702 000001              MOV      #1, R2      ;FIRST SPEED CODE
006264 012705 000007              MOV      #7, R5      ;LINE 7 WILL BE TESTED
006270 012767 103400 010062      MOV      #7*400-100000, RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;13 SPEEDS WILL BE TESTED
                                ;FIRST SPEED =50 BAUD,
                                ;8 BITS PER CHARACTER
006276 012700 000015              MOV      #15, R0
006302 012701 002103              MOV      #2103, R1
                                ;SELECT LINE 7
                                ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
006306 010577 007744              1$:    MOV      R5, @DHSCR
006312 010177 007744              MOV      R1, @DHLPR
                                ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
006316 012777 016364 007740      MOV      #TBUF, @DHBA
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
006324 012777 177400 007734      MOV      #-400, @DHBC

```

```

006332 012777 000200 007730      MOV    #200,@DHBAR      ;START TRANSMITTER
006340 105777 007712      2$:   TSTB   @DHSCR      ;WAIT FOR DATA TO BE RECEIVED
006344 100375                BPL    2$
006346 017703 007706      MOV    @DHNRC,R3      ;GET RECEIVED DATA
006352 C20367 010002      CMP    R3,RDATA      ;COMPER EXPECTED AND RECEIVED DATA
006356 001407                BEQ    3$
006360 005077 007704      CLR    @DHBAR        ;STOP TRANSMITTER
006364                HLT    1              ;DATA ERROR
006364 104001                EMT    1
006366 104410                SCOPE1
006370 012777 000200 007672      MOV    #200,@DHBAR      ;CHECK FOR LOOP AT CURRENT SPEED
006376 105267 007756      3$:   INCB   RDATA      ;RESTART TRANSMITTER
006402 001356                BNE    2$              ;UPDATA EXPECTED DATA
006404 062701 002100      ADD    #2100,R1        ;UPDATE LINE SPEED
006410 005202                INC    R2              ;UPDATE SPEED CODE
006412 005300                DEC    R0
006414 001334                BNE    1$
006416 104400      4$:   SCOPE
      000010      LINE=LINE+1
      000400      BITX=BITX+BITX
006420      SDATA2 \LINE.\BITX

      ;SINGLE LINE DATA TEST
      ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTFRS ON LINE 10
      ;CHARATER LENGTH IS 8 BITS
      ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
      ;TC 9600 BAUD.
      ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
      ;AT EACH SPEED

006420      TS \XN,1,4$,1$
006420 012767 000340 171350      T31:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
006426 012767 000001 007670      MOV    #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
006434 012767 006614 007656      MOV    #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <1$>
006442 012767 006504 007652      MOV    #1$,FREEZ1     ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
006450 012777 004000 007600      MOV    #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
006456 012702 000001                MOV    #1,R2           ;FIRST SPEED CODE
006462 012705 000010                MOV    #10,R5          ;LINE 10 WILL BE TESTED
006466 012767 104000 007664      MOV    #10*400+100000,RDATA
      ;SET EXPECTED LINE NUMBER
      ;AND VALID DATA FLAG
      ;EXPECTED DATA
      ;13 SPEEDS WILL BE TESTED
      ;FIRST SPEED =50 BAUD,
      ;8 BITS PER CHARACTER
006474 012700 000015                MOV    #15,R0          ;SELECT LINE 10
006500 012701 002103                MOV    #2103,R1       ;SET LINE SPEED AND
      ;CHARACTER LENGTH
006504 010577 007546      1$:   MOV    R5,@DHSCR     ;ADDRESS OF TRANSMITTER
006510 010177 007546      MOV    R1,@DHLPR      ;DATA BUFFER
006514 012777 016364 007542      MOV    #TBUF,@DHBA    ;400 (OCTAL) BYTES
006522 012777 177400 007536      MOV    #-400,@DHBC    ;WILL BE TRANSMITTED
006530 012777 000400 007532      MOV    #400,@DHBAR    ;START TRANSMITTER
006536 105777 007514      2$:   TSTB   @DHSCR      ;WAIT FOR DATA TO BE RECEIVED

```

```

006542 100375          BPL      2$
006544 017703 007510  MOV     @DHNRC,R3      ;GET RECEIVED DATA
006550 020367 007604  CMP     R3,RDATA      ;COMPER EXPECTED AND RECEIVED DATA
006554 001407          BEQ     3$
006556 005077 007506  CLR     @DHBAR        ;STOP TRANSMITTER
006562          HLT     1        ;DATA ERROR
006562 104001          EMT     1
006564 104410          SCOPE1
006566 012777 000400 007474  MOV     #400,@DHBAR    ;CHECK FOR LOOP AT CURRENT SPEED
006574 105267 007560 3$:     INCB   RDATA      ;RESTART TRANSMITTER
006600 001356          BNE     2$            ;UPDATA EXPECTED DATA
006602 062701 002100  ADD     @2100,R1      ;UPDATE LINE SPEED
006606 005202          INC     R2            ;UPDATE SPEED CODE
006610 005300          DEC     R0
006612 001334          BNE     1$
006614 104400          4$:     SCOPE
          000011      LINE=LINE+1
          001000      BITX=BITX+BITX
006616          SDATA2 \LINE,\BITX

          ;SINGLE LINE DATA TEST
          ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 11
          ;CHARATER LENGTH IS 8 BITS
          ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
          ;TO 9600 BAUD.
          ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
          ;AT EACH SPEED

006616          TS \XN,1,4$,1$
006616 012767 006340 171152 T32:   MOV     #340,PS      ;DISABLE ALL INTERRUPTS
006624 012767 000001 007472  MOV     #1,ICOUNT     ;SET UP FOR 1 ITERATIONS
006632 012767 007012 007460  MOV     #4$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <1$>
006640 012767 006702 007454  MOV     #1$,FREEZ1    ;SET UP TO LOOP WITH DATA      : 3
          .ENDC
          XN=XN+1
006646 012777 004000 007402  MOV     @BIT11,@DHSCR ;MASTER CLEAR INTERFACE
006654 012702 000001          MOV     #1,R2        ;FIRST SPEED CODE
006660 012705 000011          MOV     #11,R5       ;LINE 11 WILL BE TESTED
006664 012767 104400 007466  MOV     #11*400+100000,RDATA
          ;SET EXPECTED LINE NUMBER
          ;AND VALID DATA FLAG
          ;EXPECTED DATA
          ;13 SPEEDS WILL BE TESTED
          ;FIRST SPEED =50 BAUD,
          ;8 BITS PER CHARACTER
006672 012700 000015          MOV     #15,R0
006676 012701 002103          MOV     #2103,R1     ;SELECT LINE 11
          ;SET LINE SPEED AND
          ;CHARACTER LENGTH
006702 010577 007350 1$:     MOV     R5,@DHSCR    ;ADDRESS OF TRANSMITTER
006706 010177 007350          MOV     R1,@DHLPD   ;DATA BUFFER
          ;400 (OCTAL) BYTES
          ;WILL BE TRANSMITTED
          ;START TRANSMITTER
006712 012777 016364 007344  MOV     @TBUF,@DHBA ;WAIT FOR DATA TO BE RECEIVED
006720 012777 177400 007340  MOV     #-400,@DHBC
          ;SET EXPECTED LINE NUMBER
          ;AND VALID DATA FLAG
          ;EXPECTED DATA
          ;13 SPEEDS WILL BE TESTED
          ;FIRST SPEED =50 BAUD,
          ;8 BITS PER CHARACTER
006726 012777 001000 007334  MOV     #1000,@DHBAR ;START TRANSMITTER
006734 105777 007316 2$:     TSTB   @DHSCR      ;WAIT FOR DATA TO BE RECEIVED
006740 100375          BPL     2$
006742 017703 007312  MOV     @DHNRC,R3    ;GET RECEIVED DATA

```

```

006746 020367 007406      CMP      R3,RDATA      ;COMPER EXPECTED AND RECEIVED DATA
006752 001407              BEQ      3$
006754 005077 007310      CLR      @DHBAR        ;STOP TRANSMITTER
006760              HLT      1          ;DATA ERROR
006760 104001              EMT      1
006762 104410              SCOPE1
006764 012777 001000 007276  MOV      #1000,@DHBAR  ;CHECK FOR LOOP AT CURREN' SPEED
006772 105267 007362      3$:      INCB     RDATA    ;RESTART TRANSMITTER
006776 001356              BNE     2$            ;UPDATA EXPECTED DATA
007000 062701 002100      ADD     #2100,R1      ;UPDATE LINE SPEED
007004 005202              INC     R2            ;UPDATE SPEED CODE
007006 005300              DEC     R0
007010 001334              BNE     1$
007012 104400      4$:      SCOPE
          000012      LINE=LINE+1
          002000      BITX=BITX+BITX
007014      SDATA2 \LINE,\BITX

          ;SINGLE LINE DATA TEST
          ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 12
          ;CHARATER LENGTH IS 8 BITS
          ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
          ;TO 9600 BAUD.
          ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
          ;AT EACH SPEED

007014 012767 000340 170754  TS \XN,1,4$,1$
007014 012767 000001 007274  T33:    MOV     #340,PS      ;DISABLE ALL INTERRUPTS
007022 012767 000001 007274      MOV     #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
007030 012767 007210 007262      MOV     #4$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <1$>
007036 012767 007100 007256      MOV     #1$,FREEZ1    ;SET UP TO LOOP WITH DATA      ; 3
          .ENDC
          XN=XN+1
007044 012777 004000 007204      MOV     #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
007052 012702 000001              MOV     #1,R2          ;FIRST SPEED CODE
007056 012705 000012              MOV     #12,R5         ;LINE 12 WILL BE TESTED
007062 012767 105000 007270      MOV     #12*400+100000,RDATA
          ;SET EXPECTED LINE NUMBER
          ;AND VALID DATA FLAG
          ;EXPECTED DATA
          ;13 SPEEDS WILL BE TESTED
          ;FIRST SPEED =50 BAUD,
          ;8 BITS PER CHARACTER
007070 012700 000015              MOV     #15,R0
007074 012701 002103              MOV     #2103,R1
          ;SELECT LINE 12
          ;SET LINE SPEED AND
          ;CHARACTER LENGTH
007100 010577 007152      1$:      MOV     R5,@DHSCR     ;ADDRESS OF TRANSMITTER
007104 010177 007152      MOV     R1,@DHLPR    ;DATA BUFFER
          ;400 (OCTAL) BYTES
          ;WILL BE TRANSMITTED
          ;START TRANSMITTER
          ;WAIT FOR DATA TO BE RECEIVED
007110 012777 016364 007146      MOV     #TBUF,@DHBA
007116 012777 177400 007142      MOV     #-400,@DHBC
          ;GET RECEIVED DATA
          ;COMPER EXPECTED AND RECEIVED DATA
007124 012777 002000 007136      MOV     #2000,@DHBAR
007132 105777 007120      2$:      TSTB   @DHSCR
007136 100375              BPL     2$
007140 017703 007114              MOV     @DHNRC,R3
007144 020367 007210      CMP     R3,RDATA
007150 001407              BEQ     3$

```

```

007152 005077 007112          CLR    @DHBAR          ;STOP TRANSMITTER
007156                                HLT    1                ;DATA ERROR
007156 104001          EMT    1
007160 104410          SCOPE1
007162 012777 002000 007100    MOV    #2000,@DHBAR    ;CHECK FOR LOOP AT CURRENT SPEED
007170 105267 007164          3$: INCB  RDATA          ;RESTART TRANSMITTER
007174 001356          BNE    2$                ;UPDATA EXPECTED DATA
007176 062701 002100          ADD    #2100,R1        ;UPDATE LINE SPEED
007202 005202          INC    R2                ;UPDATE SPEED CODE
007204 005300          DEC    R0
007206 001334          BNE    1$
007210 104400          4$: SCOPE
      000013          LINE=LINE+1
      004000          BITX=BITX+BITX
007212                                SDATA2 \LINE,\BITX

      ;SINGLE LINE DATA TEST
      ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 13
      ;CHARATER LENGTH IS 8 BITS
      ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
      ;TO 9600 RAUD.
      ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
      ;AT EACH SPEED

007212                                TS \XN,1,4$,1$
007212 012767 000340 170556    T34: MOV    #340,PS          ;DISABLE ALL INTERRUPTS
007220 012767 000001 007076    MOV    #1,ICOUNT        ;SET UP FOR 1 ITERATIONS
007226 012767 007406 007064    MOV    #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
007234 012767 007276 007060    MOV    #1$,FREEZ1      ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
007242 012777 004000 007006    MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
007250 012702 000001          MOV    #1,R2            ;FIRST SPEED CODE
007254 012705 000013          MOV    #13,R5           ;LINE 13 WILL BE TESTED
007260 012767 105400 007072    MOV    #13*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;13 SPEEDS WILL BE TESTED
                                ;FIRST SPEED =50 BAUD,
                                ;8 BITS PER CHARACTER
007266 012700 000015          MOV    #15,R0           ;SELECT LINE 13
007272 012701 002103          MOV    #2103,R1        ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
007276 010577 006754          1$: MOV    R5,@DHSCR      ;ADDRESS OF TRANSMITTER
007302 010177 006754          MOV    R1,@DHLPR       ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
007306 012777 016364 006750    MOV    #TBUF,@DH8A     ;START TRANSMITTER
007314 012777 177400 006744    MOV    #-400,@DH8C     ;WAIT FOR DATA TO BE RECEIVED
                                ;GET RECEIVED DATA
                                ;COMPER EXPECTED AND RECEIVED DATA
007322 012777 004000 006740    MOV    #4000,@DHBAR
007330 105777 006722          2$: TSTB  @DHSCR
007334 100375          BPL    2$
007336 017703 006716          MOV    @DHNRC,R3
007342 020367 007012          CMP    R3,RDATA
007346 001407          BEQ    3$
007350 005077 006714          CLR    @DHBAR
007354                                HLT    1                ;STOP TRANSMITTER
                                ;DATA ERROR

```

```

007354 104001          EMT      1
007356 104410          SCOPE1
007360 012777 004000 006702 3$:  MOV    #4000,@DHBAR      ;CHECK FOR LOOP AT CURRENT SPEED
007366 105267 006766          INCB   RDATA      ;RESTART TRANSMITTER
007372 001356          BNE    2$      ;UPDATE EXPECTED DATA
007374 062701 002100          ADD    #2100,R1      ;UPDATE LINE SPEED
007400 005202          INC    R2      ;UPDATE SPEED CODE
007402 005300          DEC    R0
007404 001334          BNE    1$
007406 104400          4$:  SCOPE
      000014          LINE=LINE+1
      010000          BITX=BITX+BITX
007410          SDATA2 \LINE,\BITX

      ;SINGLE LINE DATA TEST
      ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 14
      ;CHARATER LENGTH IS 8 BITS
      ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
      ;TO 9600 BAUD.
      ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
      ;AT EACH SPEED

007410          7S \XN,1,4$,1$
007410 012767 000340 170360 T35:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
007416 012767 000001 006700          MOV    #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
007424 012767 007604 006666          MOV    #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
      .IF NB <1$>
007432 012767 007474 006662          MOV    #1$,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1
007440 012777 004000 006610          MOV    #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
007446 012702 000001          MOV    #1,R2      ;FIRST SPEED CODE
007452 012705 000014          MOV    #14,R5      ;LINE 14 WILL BE TESTED
007456 012767 106000 006674          MOV    #14*400+100000,RDATA
      ;SET EXPECTED LINE NUMBER
      ;AND VALID DATA FLAG
      ;EXPECTED DATA
      ;13 SPEEDS WILL BE TESTED
      ;FIRST SPEED =50 BAUD,
      ;8 BITS PER CHARACTER
      ;SELECT LINE 14
      ;SET LINE SPEED AND
      ;CHARACTER LENGTH
      ;ADDRESS OF TRANSMITTER
      ;DATA BUFFER
      ;400 (OCTAL) BYTES
      ;WILL BE TRANSMITTED
      ;START TRANSMITTER
      ;WAIT FOR DATA TO BE RECEIVED

007464 012700 000015          MOV    #15,R0
007470 012701 002103          MOV    #2103,R1
      ;GET RECEIVED DATA
      ;COMPER EXPECTED AND RECEIVED DATA

007474 010577 006556          1$:  MOV    R5,@DHSCR
007500 010177 006556          MOV    R1,@DHLPR
      ;STOP TRANSMITTER
      ;DATA ERROR

007504 012777 016364 006552          MOV    #1BUF,@DHBA
      ;CHECK FOR LOOP AT CURRENT SPEED

007512 012777 177400 006546          MOV    #-400,@DHBC
      ;CHECK FOR LOOP AT CURRENT SPEED

007520 012777 010000 006542          MOV    #10000,@DHBAR
007526 105777 006524          2$:  TSTB  @DHSCR
007532 100375          BPL    2$
007534 017703 006520          MOV    @DHNRC,R3
007540 020367 006614          CMP    R3,RDATA
007544 001407          BEQ    3$
007546 005077 006516          CLR    @DHBAR
007552          HLT    1
007552 104001          EMT    1
007554 104410          SCOPE1

```



```

007556 012777 010000 006504      MOV    #10000,@DHBAR      ;RESTART TRANSMITTER
007564 105267 006570      3$:   INCB   RDATA        ;UPDATA EXPECTED DATA
007570 001356      BNE    2$
007572 062701 002100      ADD    #2100,R1        ;UPDATE LINE SPEED
007576 005202      INC    R2              ;UPDATE SPEED CODE
007600 005300      DEC    R0
007602 001334      BNE    1$
007604 104400      4$:   SCOPE
      000015      LINE=LINE+1
      020000      BITX=BITX+BITX
007606      SDATA2 \LINE,\BITX

      ;SINGLE LINE DATA TEST
      ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 15
      ;CHARATER LENGTH IS 8 BITS
      ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
      ;TO 9600 BAUD.
      ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
      ;AT EACH SPEED

007606      TS \XN,1,4$,1$
007606 012767 000340 170162      T36:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
007614 012767 000001 006502      MOV    #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
007622 012767 010002 006470      MOV    #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

007630 012767 007672 006464      .IF NB <1$>
      MOV    #1$,FREEZ1  ;SET UP TO LOOP WITH DATA      ; 3
      .ENDC
      XN=XN+1

007636 012777 004000 006412      MOV    #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
007644 012702 000001      MOV    #1,R2          ;FIRST SPEED CODE
007650 012705 000015      MOV    #15,R5         ;LINE 15 WILL BE TESTED
007654 012767 106400 006476      MOV    #15*400+100000,RDATA
      ;SET EXPECTED LINE NUMBER
      ;AND VALID DATA FLAG
      ;EXPECTED DATA
      ;13 SPEEDS WILL BE TESTED
      ;FIRST SPEED =50 BAUD,
      ;8 BITS PER CHARACTER
      ;SELECT LINE 15
      ;SET LINE SPEED AND
      ;CHARACTER LENGTH
      ;ADDRESS OF TRANSMITTER
      ;DATA BUFFER
      ;400 (OCTAL) BYTES
      ;WILL BE TRANSMITTED
      ;START TRANSMITTER
      ;WAIT FOR DATA TO BE RECEIVED

007662 012700 000015      MOV    #15,R0
007666 012701 002103      MOV    #2103,R1      ;8 BITS PER CHARACTER

007672 010577 006360      1$:   MOV    R5,@DHSCR     ;SELECT LINE 15
007676 010177 006360      MOV    R1,@DHLPR     ;SET LINE SPEED AND
      ;CHARACTER LENGTH
      ;ADDRESS OF TRANSMITTER
      ;DATA BUFFER
      ;400 (OCTAL) BYTES
      ;WILL BE TRANSMITTED
      ;START TRANSMITTER
      ;WAIT FOR DATA TO BE RECEIVED

007702 012777 016364 006354      MOV    #TBUF,@DHBA
      ;ADDRESS OF TRANSMITTER
      ;DATA BUFFER
      ;400 (OCTAL) BYTES
      ;WILL BE TRANSMITTED
      ;START TRANSMITTER
      ;WAIT FOR DATA TO BE RECEIVED

007710 012777 177400 006350      MOV    #-400,@DHBC
      ;400 (OCTAL) BYTES
      ;WILL BE TRANSMITTED
      ;START TRANSMITTER
      ;WAIT FOR DATA TO BE RECEIVED

007716 012777 020000 006344      MOV    #20000,@DHBAR
007724 105777 006326      2$:   TSTB   @DHSCR
007730 100375      BPL    2$
007732 017703 006322      MOV    @DHNR,R3
007736 020367 006416      CMP    R3,RDATA      ;GET RECEIVED DATA
      ;COMPER EXPECTED AND RECEIVED DATA
007742 001407      BEQ    3$
007744 005077 006320      CLR    @DHBAR
007750      HLT    1            ;STOP TRANSMITTER
      ;DATA ERROR
007750 104001      EMT    1
007752 104410      SCOPE1
007754 012777 020000 006306      MOV    #20000,@DHBAR
007762 105267 006372      3$:   INCB   RDATA        ;CHECK FOR LOOP AT CURRENT SPEED
      ;RESTART TRANSMITTER
      ;UPDATA EXPECTED DATA

```

```

007766 001356          BNE      2$
007770 062701 002100  ADD      #2100,R1      ;UPDATE LINE SPEED
007774 005202          INC      R2             ;UPDATE SPEED CODE
007776 005300          DEC      R0
010000 001334          BNE      1$
010002 104400          4$: SCOPE
          LINE=LINE+1
          BITX=BITX+BITX
          SDATA2 \LINE,\BITX

          ;SINGLE LINE DATA TEST
          ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 16
          ;CHARATER LENGTH IS 8 BITS
          ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
          ;TO 9600 BAUD.
          ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
          ;AT EACH SPEED

010004          TS \XN,1,4$,1$
010004 012767 000340 167764 T37:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
010012 012767 000001 006304      MOV      #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
010020 012767 010200 006272      MOV      #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

          .IF NB <1$>
010026 012767 010070 006266      MOV      #1$,FREEZ1     ;SET UP TO LOOP WITH DATA      : 3
          .ENDC
          XN=XN+1
010034 012777 004000 006214      MOV      #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
010042 012702 000001          MOV      #1,R2          ;FIRST SPEED CODE
010046 012705 000016          MOV      #16,R5         ;LINE 16 WILL BE TESTED
010052 012767 107000 006300      MOV      #16*400+100000,RDATA
          ;SET EXPECTED LINE NUMBER
          ;AND VALID DATA FLAG
          ;EXPECTED DATA
          ;13 SPEEDS WILL BE TESTED
          ;FIRST SPEED =50 BAUD,
          ;8 BITS PER CHARACTER
010060 012700 000015          MOV      #15,R0
010064 012701 002103          MOV      #2103,R1      ;SELECT LINE 16
          ;SET LINE SPEED AND
          ;CHARACTER LENGTH
010070 010577 006162          1$: MOV      R5,@DHSCR
010074 010177 006162          MOV      R1,@DHLPR     ;SET LINE SPEED AND
          ;CHARACTER LENGTH
010100 012777 016364 006156      MOV      #TBUF,@DHBA   ;ADDRESS OF TRANSMITTER
          ;DATA BUFFER
010106 012777 177400 006152      MOV      #-400,@DHBC   ;400 (OCTAL) BYTES
          ;WILL BE TRANSMITTED
          ;START TRANSMITTER
010114 012777 040000 006146      MOV      #40000,@DHBAR ;WAIT FOR DATA TO BE RECEIVED
010122 105777 006130          2$: TSTB    @DHSCR
010126 100375          BPL      2$
010130 017703 006124          MOV      @DHNR,R3      ;GET RECEIVED DATA
010134 020367 006220          CMP      R3,RDATA     ;COMPER EXPECTED AND RECEIVED DATA
010140 001407          BEQ     3$
010142 005077 006122          CLR     @DHBAR
010146          HLT     1      ;STOP TRANSMITTER
          ;DATA ERROR
010146 104001          EMT     1
010150 104410          SCOPE1
010152 012777 040000 006110      MOV      #40000,@DHBAR ;CHECK FOR LOOP AT CURRENT SPEED
010160 105267 006174          3$: INCB    RDATA        ;RESTART TRANSMITTER
010164 001356          BNE     2$             ;UPDATA EXPECTED DATA
010166 062701 002100          ADD     #2100,R1      ;UPDATE LINE SPEED

```

```

010172 005202          INC      R2          ;UPDATE SPEED CODE
010174 005300          DEC      R0
010176 001334          BNE     1$
010200 104400          4$: SCOPE
          000017        LINE=LINE+1
          100000        BITX=BITX+BITX
010202          SDATA2 \LINE,\BITX

          ;SINGLE LINE DATA TEST
          ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 17
          ;CHARATER LENGTH IS 8 BITS
          ;LINE SPEED WILL START AT 50 BAUD AND BE INCREMENTED
          ;TO 9600 BAUD.
          ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
          ;AT EACH SPEED

010202          TS \XN,1,4$,1'
010202 012767 000340 167566 T40: MOV     #340,PS          ;DISABLE ALL INTERRUPTS
010210 012767 000001 006106     MOV     #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
010216 012767 010376 006074     MOV     #4$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
          .IF NB <1$>
010224 012767 010266 006070     MOV     #1$,FREEZ1    ;SET UP TO LOOP WITH DATA          ; 3
          .ENDC
          XN=XN+1
010232 000041          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
010240 012702 004000 006016     MOV     #1,R2         ;FIRST SPEED CODE
010244 012705 000017          MOV     #17,R5        ;LINE 17 WILL BE TESTED
010250 012767 107400 006102     MOV     #17*400+100000,RDATA
          ;SET EXPECTED LINE NUMBER
          ;AND VALID DATA FLAG
          ;EXPECTED DATA
          ;13 SPEEDS WILL BE TESTED
          ;FIRST SPEED =50 BAUD.
          ;8 BITS PER CHARACTER
010256 012700 000015          MOV     #15,R0        ;SELECT LINE 17
010262 012701 002103          MOV     #2103,R1     ;SET LINE SPEED AND
          ;CHARACTER LENGTH
010266 010577 005764          1$: MOV     R5,@DHSCR   ;ADDRESS OF TRANSMITTER
010272 010177 005764          MOV     R1,@DHLPR   ;DATA BUFFER
          ;400 (OCTAL) BYTES
          ;WILL BE TRANSMITTED
          ;START TRANSMITTER
          ;WAIT FOR DATA TO BE RECEIVED
010276 012777 016364 005760     MOV     #TBUF,@DHBA
010304 012777 177400 005754     MOV     #-400,@DHBC
          ;GET RECEIVED DATA
          ;COMPER EXPECTED AND RECEIVED DATA
010312 012777 100000 005750     MOV     #100000,@DHBAR
010320 105777 005732          2$: TSTB   @DHSCR
010324 100375          BPL     2$
010326 017703 005726          MOV     @DHNRC,R3
010332 020367 006022          CMP     R3,RDATA
010336 001407          BEQ     3$
010340 005077 005724          CLR     @DHBAR
          ;STOP TRANSMITTER
          ;DATA ERROR
010344          HLT     1
010344 104001          EMT     1
010346 104410          SCOPE1
010350 012777 100000 005712     MOV     #100000,@DHBAR
          ;CHECK FOR LOOP AT CURRENT SPEED
          ;RESTART TRANSMITTER
          ;UPDATA EXPECTED DATA
010356 105267 005776          3$: INCB   RDATA
010362 001356          BNE     2$
010364 062701 002100          ADD     #2100,R1
          ;UPDATE LINE SPEED
          ;UPDATE SPEED CODE
010370 005202          INC     R2
010372 005300          DEC     R0

```

```

010374 001334          BNE      1$
010376 104400          4$:    SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
30      000020          LINE=0
31      000000          XLINE=LINE
32      000001          BITX=1
33      000001          XBIT=BITX
35      000020          .REPT   20
36      000000          SDATA3  \LINE,\BITX
37      000000          .NLIST
38      000001          LINE=LINE+1
39      000001          BITX=BITX+BITX
40      000000          .LIST
41      000000          .ENDR
010400          SDATA3  \LINE,\BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 0
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

010400          TS \XN,1,4$,1$
010400 012767 000340 167370 T41:  MOV      #340,PS          ;DISABLE ALL INTERRUPTS
010406 012767 000001 005710      MOV      #1,ICOUNT        ;SET UP FOR 1 ITERATIONS
010414 012767 010616 005676      MOV      #4$,ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
010422 012767 010470 005672      MOV      #1$,FREEZ1        ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
010430 012777 004000 005620      MOV      #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
010436 005004          CLR      R4                  ;FIRST CHARACTER LENGTH CODE (5 BITS)
010440 012705 000000          MOV      #0,R5              ;LINE 0 WILL BE TESTED
010444 012767 100000 005706      MOV      #0*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4 CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS,,
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
                                ;SELECT LINE 0

010452 012700 000004          MOV      #4,R0
010456 012701 033500          MOV      #33500,R1
                                ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
                                ;START TRANSMITTER
                                ;WAIT FOR DATA TO BE RECEIVED
010462 012767 177740 005672      MOV      #-40,BYTCNT
010470 010577 005562          1$:  MOV      R5,@DHSCR
010474 016702 005662          MOV      BYTCNT,R2
010500 005402          NEG      R2
010502 010177 005554          MOV      R1,@DHLPR

010506 012777 016364 005550          MOV      #TBUF,@DHBA
010514 016777 005642 005544          MOV      BYTCNT,@DHBC
010522 012777 000001 005540          MOV      #1,@DHBAR
010530 105777 005522          2$:  TSTB   @DHSCR
010534 100375          BPL     2$

```

```

010536 017703 005516      MOV    @DHNRC,R3      ;GET RECEIVED DATA
010542 020367 005612      CMP    R3,RDATA      ;COMPER EXPECTED AND RECEIVED DATA
010546 001407              BEQ    3$             ;
010550 005077 005514      CLR    @DHBAR         ;STOP TRANSMITTER
010554              HLT    2             ;DATA ERROR
010554 104002              EMT    2             ;
010556 104410              SCOPE1              ;CHECK FOR LOOP AT CURRENT SPLEED
010560 012777 000001 005502 3$: MOV    @1,@DHBAR      ;RESTART TRANSMITTER
010566 105267 005566      INCB   RDATA         ;UPDATA EXPECTED DATA
010572 005302              DEC    R2             ;
010574 001355              BNE   2$             ;
010576 105067 005556      CLRB  RDATA         ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH
010602 005201              INC    R1             ;
010604 005204              INC    R4             ;
010606 006367 005550      ASL   BYTCNT         ;
010612 005300              DEC    R0             ;
010614 001325              BNE   1$             ;
010616 104400              4$:  SCOPE          ;
                                LINE=LINE+1
                                BITX=BITX+BITX
                                SDATA3 \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 1
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH
010620              TS \XN,1,4$,1$
010620 012767 000340 167150 T42: MOV    @340,PS      ;DISABLE ALL INTERRUPTS
010626 012767 000001 005470      MOV    @1,ICOUNT     ;SET UP FOR 1 ITERATIONS
010634 012767 011036 005456      MOV    @4$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
010642 012767 010710 005452      MOV    @1$,FREEZ1    ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
010650 012777 004000 005400      MOV    @BIT11,@DHSCR ;MASTER CLEAR INTERFACE
010656 005004              CLR    R4             ;FIRST CHARACTER LENGTH CODE (5 BITS)
010660 012705 000001              MOV    @1,R5          ;LINE 1 WILL BE TESTED
010664 012767 100400 005466      MOV    @1*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS.,.
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
                                ;SELECT LINE 1
010672 012700 000004              MOV    @4,R0         ;
010676 012701 033500              MOV    @33500,R1     ;
010702 012767 177740 005452      MOV    @-40,BYTCNT   ;
010710 010577 005342              1$: MOV    R5,@DHSCR    ;
010714 016702 005442              MOV    BYTCNT,R2     ;
010720 005402              NEG    R2             ;
010722 010177 005334              MOV    R1,@DHLPR     ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
010726 012777 016364 005330      MOV    @TBUF,@DHBA

```

```

010734 016777 005422 005324      MOV      BYTCNT,@DHBC      ;DATA BUFFER
                                ;400 (OCTAL) BYTES
010742 012777 000002 005320      MOV      #2,@DHBAR        ;WILL BE TRANSMITTED
010750 105777 005302              2$:  TSTB     @DHSCR          ;START TRANSMITTER
010754 100375                    BPL      2$                ;WAIT FOR DATA TO BE RECEIVED
010756 017703 005276      MOV      @DHNRC,R3        ;GET RECEIVED DATA
010762 020367 005372      CMP      R3,RDATA        ;COMPER EXPECTED AND RECEIVED DATA
010766 001407                    BEQ      3$
010770 005077 005274      CLR      @DHBAR          ;STOP TRANSMITTER
010774                    HLT      2                        ;DATA ERROR
                                EMT      2
010776 104410                    SCOPE1
011000 012777 000002 005262      MOV      #2,@DHBAR        ;CHECK FOR LOOP AT CURRENT SPEED
011006 105267 005346              3$:  INCB     RDATA          ;RESTART TRANSMITTER
011012 005302                    DEC      R2                ;UPDATA EXPECTED DATA
011014 001355                    BNE     2$
011016 105067 005336      CLRB     RDATA          ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH
011022 005201                    INC      R1
011024 005204                    INC      R4
011026 006367 005330      ASL     BYTCNT
011032 005300                    DEC      R0
011034 001325                    BNE     1$
011036 104400                    4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
011040 000002                    SDATA3  \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 2
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

011040                    TS \XN,1,4$,1$
011040 012767 000340 166730      T43:  MOV      #340,PS      ;DISABLE ALL INTERRUPTS
011046 012767 000001 005250      MOV      #1,ICOUNT        ;SET UP FOR 1 ITERATIONS
011054 012767 011256 005236      MOV      #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
011062 012767 011130 005232      MOV      #1$,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
011070 012777 004000 005160      MOV      @BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
011076 005004                    CLR      R4                ;FIRST CHARACTER LENGTH CODE (5 BITS)
011100 012705 000002                    MOV      #2,R5            ;LINE 2 WILL BE TESTED
011104 012767 101000 005246      MOV      #2*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4CHARACTER LENGTHS
                                ;WILL BE TESTED
011112 012700 000004                    MOV      #4,R0            ;FIRST CHARACTER LENGTH =5 BITS..
011116 012701 033500                    MOV      #33500,R1        ;LINE SPEED =9600 BAUD
011122 012767 177740 005232      MOV      # 40,BYTCNT     ;40 CHARACTERS AT 5 BITS

```

```

011130 010577 005122      1$:  MOV    R5,@DHSCR      ;SELECT LINE 2
011134 016702 005222      MOV    BYTCNT,R2
011140 005402              NEG    R2
011142 010177 005114      MOV    R1,@DHLPR      ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
011146 012777 016364 005110  MOV    @TBUF,@DHBA      ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
011154 016777 005202 005104  MOV    BYTCNT,@DHBC      ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
011162 012777 000004 005100  MOV    #4,@DHBAR        ;START TRANSMITTER
011170 105777 005062      2$:  TSTB   @DHSCR            ;WAIT FOR DATA TO BE RECEIVED
011174 100375              BPL    @DHSCR            2$
011176 017703 005056      MOV    @DHNRC,R3        ;GET RECEIVED DATA
011202 020367 005152      CMP    R3,RDATA         ;COMPER EXPECTED AND RECEIVED DATA
011206 001407              BEQ    3$
011210 005077 005054      CLR    @DHBA           ;STOP TRANSMITTER
011214              HLT    2                ;DATA ERROR
011214 104002              EMT    2
011216 104410              SCOPE1
011220 012777 000004 005042  MOV    #4,@DHBAR        ;CHECK FOR LOOP AT CURRENT SPEED
011226 105267 005126      3$:  INCB   RDATA           ;RESTART TRANSMITTER
011232 005302              DEC    R2                ;UPDATA EXPECTED DATA
011234 001355              BNE   2$
011236 105067 005116      CLRB  RDATA            ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
011242 005201              INC    R1                ;UPDATA CHARACTER LENGTH
011244 005204              INC    R4
011246 006367 005110      ASL   BYTCNT
011252 005300              DEC    R0
011254 001325              BNE   1$
011256 104400      4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
011260      SDATA3 \LINE,\BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 3
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

011260      TS \XN,1,4$,1$
011260 012767 000340 166510  T44:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
011266 012767 000001 005030      MOV    #1,ICOUNT       ;SET UP FOR 1 ITERATIONS
011274 012767 011476 005016      MOV    #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
011302 012767 011350 005012      MOV    #1$,FREEZ1      ;SET UP TO LOOP WITH DATA      : 3
                                .ENDC
                                XN=XN+1
011310 012777 004000 004740      MOV    #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
011316 005004              CLR    R4                ;FIRST CHARACTER LENGTH CODE (5 BITS)
011320 012705 000003              MOV    #3,R5             ;LINE 3 WILL BE TESTED
011324 012767 101400 005026      MOV    #3*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG

```

```

011332 012700 000004          MOV    #4,R0          ;EXPECTED DATA
                                ;4CHARACTER LENGTHS
011336 012701 033500          MOV    #33500,R1      ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS,,
011342 012767 177740 005012    MOV    #-40,BYTCNT   ;LINE SPEED =9600 BAUD
011350 010577 004702          1$:  MOV    R5,@DHSCR  ;40 CHARACTERS AT 5 BITS
011354 016702 005002          MOV    BYTCNT,R2    ;SELECT LINE 3
011360 005402                NEG    R2
011362 010177 004674          MOV    R1,@DHLPR    ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
011366 012777 016364 004670    MOV    #TBUF,@DHBA  ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
011374 016777 004762 004664    MOV    BYTCNT,@DHBC ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
011402 012777 000010 004660    MOV    #10,@DHBAR   ;START TRANSMITTER
011410 105777 004642          2$:  TSTB   @DHSCR      ;WAIT FOR DATA TO BE RECEIVED
011414 100375                BPL    2$
011416 017703 004636          MOV    @DHNRC,R3    ;GET RECEIVED DATA
011422 020367 004732          CMP    R3,RDATA    ;COMPER EXPECTED AND RECEIVED DATA
011426 001407                BEQ    3$
011430 005077 004634          CLR    @DHBAR      ;STOP TRANSMITTER
011434                HLT    2          ;DATA ERROR
011434 104002                EMT    2
011436 104410                SCOPE1
011440 012777 000010 004622    MOV    #10,@DHBAR   ;CHECK FOR LOOP AT CURRENT SPEED
011446 105267 004706          3$:  INCB   RDATA      ;RESTART TRANSMITTER
011452 005302                DEC    R2          ;UPDATA EXPECTED DATA
011454 001355                BNE   2$
011456 105067 004670          CLRB  RDATA        ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH
011462 005201                INC    R1
011464 005204                INC    R4
011466 006367 004670          ASL   BYTCNT
011472 005300                DEC    R0
011474 001325                BNE   1$
011476 104400          4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
011500          SDATA3 \LINE,\BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 4
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

011500          TS \XN,1,4$,1$
011500 012767 000340 166270    T45:  MOV    #340,PS  ;DISABLE ALL INTERRUPTS
011506 012767 000001 004610    MOV    #1,ICOUNT   ;SET UP FOR 1 ITERATIONS
011514 012767 011716 004576    MOV    #4$,ESCAPE  ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
011522 012767 011570 004572    MOV    #1$,FREEZ1  ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
000046

```



```

011530 012777 004000 004520      MOV    #BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
011536 005004                      CLR    R4                 ;FIRST CHARACTER LENGTH CODE (5 BITS)
011540 012705 000004                      MOV    #4,R5              ;LINE 4 WILL BE TESTED
011544 012767 102000 004606      MOV    #4*400+100000,RDATA ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4 CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS.,.
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
                                ;SELECT LINE 4

011552 012700 000004                      MOV    #4,R0

011556 012701 033500                      MOV    #33500,R1

011562 012767 177740 004572      MOV    #40,BYTCNT
011570 010577 004462      1$:  MOV    R5,@DHSCR
011574 016702 004562      MOV    BYTCNT,R2
011600 005402                      NEG    R2
011602 010177 004454                      MOV    R1,@DHLPR

                                ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
                                ;START TRANSMITTER
                                ;WAIT FOR DATA TO BE RECEIVED

011606 012777 016364 004450      MOV    #TBUF,@DHBA

011614 016777 004542 004444      MOV    BYTCNT,@DHBC

011622 012777 000020 004440      MOV    #20,@DHBAR
011630 105777 004422      2$:  TSTB  @DHSCR
011634 100375                      BPL   2$
011636 017703 004416      MOV    @DHNRC,R3
011642 020367 004512      CMP    R3,RDATA
011646 001407                      BEQ   3$
011650 005077 004414                      CLR    @DHBAR
011654                      HLT   2
011654 104002                      EMT   2
011656 104410                      SCOPE1
011660 012777 000020 004402      MOV    #20,@DHBAR
011666 105267 004466      3$:  INCB  RDATA
011672 005302                      DEC   R2
011674 001355                      BNE   2$
011676 105067 004456                      CLRB  RDATA

                                ;GET RECEIVED DATA
                                ;COMPER EXPECTED AND RECEIVED DATA

                                ;STOP TRANSMITTER
                                ;DATA ERROR

                                ;CHECK FOR LOOP AT CURRENT SPEED
                                ;RESTART TRANSMITTER
                                ;UPDATA EXPECTED DATA

011702 005201                      INC   R1
011704 005204                      INC   R4
011706 006367 004450      ASL   BYTCNT
011712 005300                      DEC   R0
011714 001325                      BNE   1$
011716 104400      4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
                                SDATA3 \LINE,\BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 5
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

011720                                TS \XN,1,4$,1$
011720 012767 000340 166050      T46:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS

```

```

011726 012767 000001 004370      MOV    #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
011734 012767 012136 004356      MOV    #4$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
011742 012767 012010 004352      MOV    #1$,FREEZ1    ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
011750 012777 004000 004300      MOV    #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
011756 005004                    CLR    R4             ;FIRST CHARACTER LENGTH CODE (5 BITS)
011760 012705 000005                    MOV    #5,R5         ;LINE 5 WILL BE TESTED
011764 012767 102400 004366      MOV    #5*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4 CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS,,
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
                                ;SELECT LINE 5
011772 012700 000004                    MOV    #4,R0
011776 012701 033500                    MOV    #33500,R1
                                ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
                                ;START TRANSMITTER
                                ;WAIT FOR DATA TO BE RECEIVED
012002 012767 177740 004352      MOV    #-40,BYTCNT
012010 010577 004242      1$:  MOV    R5,@DHSCR
012014 016702 004342      MOV    BYTCNT,R2
012020 005402                    NEG    R2
012022 010177 004234      MOV    R1,@DHLPR
                                ;GET RECEIVED DATA
                                ;COMPER EXPECTED AND RECEIVED DATA
012026 012777 016364 004230      MOV    #TBUF,@DHBA
012034 016777 004322 004224      MOV    BYTCNT,@DHBC
                                ;STOP TRANSMITTER
                                ;DATA ERROR
012042 012777 000040 004220      MOV    #40,@DHBAR
012050 105777 004202      2$:  TSTB  @DHSCR
012054 100375                    BPL   2$
012056 017703 004176      MOV    @DHNRC,R3
012062 020367 004272      CMP    R3,RDATA
012066 001407                    BEQ   3$
012070 005077 004174      CLR    @DHBAR
012074                    HLT   2
012074 104002                    EMT   2
                                ;CHECK FOR LOOP AT CURRENT SPEED
                                ;RESTART TRANSMITTER
                                ;UPDATA EXPECTED DATA
012076 104410                    SCOPE1
012100 012777 000040 004162      MOV    #40,@DHBAR
012106 105267 004246      3$:  INCB  RDATA
012112 005302                    DEC   R2
012114 001355                    BNE  2$
012116 105067 004236      CLRB  RDATA
                                ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH
012122 005201                    INC   R1
012124 005204                    INC   R4
012126 006367 004230      ASL   BYTCNT
012132 005300                    DEC   R0
012134 001325                    BNE  1$
012136 104400      4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
                                SDATA3 \LINE,\BITX
012140
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 6
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED

```


012360

SDATA3 \LINE,\BITX

```

;SINGLE LINE DATA TEST
;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 7
;LINE SPEED IS 9600 BAUD
;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
;TO 8 BITS
;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
;AT EACH CHARACTER LENGTH

```

```

012360      TS \XN,1,4$,1$
012360 012767 000340 165410 T50:  MOV    #340,PS           ;DISABLE ALL INTERRUPTS
012366 012767 000001 003730      MOV    #1,ICOUNT        ;SET UP FOR 1 ITERATIONS
012374 012767 012576 003716      MOV    #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
012402 012767 012450 003712      MOV    #1$,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
012410 012777 004000 003640      MOV    #8BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
012416 005004      CLR    R4              ;FIRST CHARACTER LENGTH CODE (5 BITS)
012420 012705 000007      MOV    #7,R5           ;LINE 7 WILL BE TESTED
012424 012767 103400 003726      MOV    #7*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4 CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS..
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
                                ;SELECT LINE 7
012432 012700 000004      MOV    #4,R0
012436 012701 033500      MOV    #33500,R1
                                ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
                                ;START TRANSMITTER
                                ;WAIT FOR DATA TO BE RECEIVED
012442 012767 177740 003712      MOV    #-40,BYTCNT
012450 010577 003602      1$:  MOV    R5,@DHSCR
012454 016702 003702      MOV    BYTCNT,R2
012460 005402      NEG    R2
012462 010177 003574      MOV    R1,@DHLPR
                                ;GET RECEIVED DATA
                                ;COMPER EXPECTED AND RECEIVED DATA
012466 012777 016364 003570      MOV    #TBUF,@DHBA
                                ;STOP TRANSMITTER
                                ;DATA ERROR
012474 016777 003662 003564      MOV    BYTCNT,@DHBC
                                ;CHECK FOR LOOP AT CURRENT SPEED
                                ;RESTART TRANSMITTER
                                ;UPDATA EXPECTED DATA
012502 012777 000200 003560      MOV    #200,@DHBAR
012510 105777 003542      2$:  TSTB  @DHSCR
012514 100375      BPL    2$
012516 017703 003536      MOV    @DHNRC,R3
012522 020367 003632      CMP    R3,RDATA
012526 001407      BEQ    3$
012530 005077 003534      CLR    @DHBAR
012534      HLT    2
012534 104002      EMT    2
012536 104410      SCOPE1
012540 012777 000200 003522      MOV    #200,@DHBAR
012546 105267 003606      3$:  INCB  RDATA
012552 005302      DEC    R2
012554 001355      BNE    2$
012556 105067 003576      CLRB  RDATA
                                ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH
012562 005201      INC    R1
012564 005204      INC    R4

```

012566 006367 003570
 012572 005300
 012574 001325
 012576 104400
 000010
 000400
 012600

ASL BYTCNT
 DEC R0
 BNE 1\$
 4\$: SCOPE
 LINE=LINE+1
 BITX=BITX+BITX
 SDATA3 \LINE,\BITX

;SINGLE LINE DATA TEST
 ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 10
 ;LINE SPEED IS 9600 BAUD
 ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
 ;TO 8 BITS
 ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
 ;AT EACH CHARACTER LENGTH

012600				TS \XN,1,4\$,1\$		
012600	012767	000340	165170	T51: MOV	#340,PS	;DISABLE ALL INTERRUPTS
012606	012767	000001	003510	MOV	#1,ICOUNT	;SET UP FOR 1 ITERATIONS
012614	012767	013016	003476	MOV	#4\$,ESCAPE	;SET UP TO ESCAPE TO NEXT TEST
				.IF NB	<1\$>	
012622	012767	012670	003472	MOV	#1\$,FREEZ1	;SET UP TO LOOP WITH DATA ; 3
				.ENDC		
				XN=XN+1		
012630	012777	004000	003420	MOV	#BIT11,@DHSCR	;MASTER CLEAR INTERFACE
012636	005004			CLR	R4	;FIRST CHARACTER LENGTH CODE (5 BITS)
012640	012705	000010		MOV	#10,R5	;LINE 10 WILL BE TESTED
012644	012767	104000	003506	MOV	#10*400+100000,RDATA	
						;SET EXPECTED LINE NUMBER
						;AND VALID DATA FLAG
						;EXPECTED DATA
012652	012700	000004		MOV	#4,R0	;4 CHARACTER LENGTHS
						;WILL BE TESTED
012656	012701	033500		MOV	#33500,R1	;FIRST CHARACTER LENGTH =5 BITS..
						;LINE SPEED =9600 BAUD
012662	012767	177740	003472	MOV	#-40,BYTCNT	;40 CHARACTERS AT 5 BITS
012670	010577	003362		1\$: MOV	R5,@DHSCR	;SELECT LINE 10
012674	016702	003462		MOV	BYTCNT,R2	
012700	005402			NEG	R2	
012702	010177	003354		MOV	R1,@DHLPR	;SET LINE SPEED AND
						;CHARACTER LENGTH
012706	012777	016364	003350	MOV	#TBUF,@DHBA	;ADDRESS OF TRANSMITTER
						;DATA BUFFER
012714	016777	003442	003344	MOV	BYTCNT,@DHBC	;400 (OCTAL) BYTES
						;WILL BE TRANSMITTED
012722	012777	000400	003340	MOV	#400,@DHBAR	;START TRANSMITTER
012730	105777	003322		2\$: TSTB	@DHSCR	;WAIT FOR DATA TO BE RECEIVED
012734	100375			BPL	2\$	
012736	017703	003316		MOV	@DHNR,R3	;GET RECEIVED DATA
012742	020367	003412		CMP	R3,RDATA	;COMPER EXPECTED AND RECEIVED DATA
012746	001407			BEQ	3\$	
012750	005077	003314		CLR	@DHBAR	;STOP TRANSMITTER
012754				HLT	2	;DATA ERROR
012754	104002			EMT	2	
012756	104410			SCOPE1		;CHECK FOR LOOP AT CURRENT SPEED
012760	012777	000400	003302	MOV	#400,@DHBAR	;RESTART TRANSMITTER
012766	105267	003366		3\$: INCB	RDATA	;UPDATA EXPECTED DATA

```

012772 005302          DEC      R2
012774 001355          BNE     2$
012776 105067 003356  CLR8    RDATA          ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH
013002 005201          INC     R1
013004 005204          INC     R4
013006 006367 003350  ASL     BYTCNT
013012 005300          DEC     R0
013014 001325          BNE     1$
013016 104400          4$: SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
013020 000011          SDATA3 \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 11
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

013020 012767 000340 164750 TS \XN,1,4$,1$
013020 012767 000001 003270 T52:  MOV     #340,PS          ;DISABLE ALL INTERRUPTS
013026 012767 000001 003270      MOV     #1,ICOUNT        ;SET UP FOR 1 ITERATIONS
013034 012767 013236 003256      MOV     #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
013042 012767 013110 003252      MOV     #1$,FREEZ1     ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
013050 012777 004000 003200      MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
013056 005004          CLR     R4              ;FIRST CHARACTER LENGTH CODE (5 BITS)
013060 012705 000011          MOV     #11,R5          ;LINE 11 WILL BE TESTED
013064 012767 104400 003266      MOV     #11*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4 CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS..
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
                                ;SELECT LINE 11

013072 012700 000004          MOV     #4,R0
013076 012701 033500          MOV     #33500,R1
                                ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
                                ;START TRANSMITTER
                                ;WAIT FOR DATA TO BE RECEIVED
013102 012767 177740 003252      MOV     #-40,BYTCNT
013110 010577 003142          1$: MOV     R5,@DHSCR
013114 016702 003242          MOV     BYTCNT,R2
013120 005402          NEG     R2
013122 010177 003134          MOV     R1,@DHLPR
                                ;GET RECEIVED DATA
                                ;COMPER EXPECTED AND RECEIVED DATA

013126 012777 016364 003130      MOV     #TBUF,@DHBA
013134 016777 003222 003124      MOV     BYTCNT,@DHBC
013142 012777 001000 003120      MOV     #1000,@DHBAR
013150 105777 003102          2$: TSTB  @DHSCR
013154 100375          BPL     2$
013156 017703 003076          MOV     @DHNRC,R3
013162 020367 003172          CMP     R3,RDATA
013166 001407          BEQ     3$

```

```

013170 005077 003074 CLR @DHBAR ;STOP TRANSMITTER
013174 HLT 2 ;DATA ERROR
013174 104002 EMT 2
013176 104410 SCOPE1 ;CHECK FOR LOOP AT CURRENT SPEED
013200 017777 001000 003062 MOV #1000,@DHBAR ;RESTART TRANSMITTER
013206 105267 003146 3$: INCB RDATA ;UPDATA EXPECTED DATA
013212 005302 DEC R2
013214 001355 BNE 2$
013216 105067 003136 CLRB RDATA ;INITIALIZE EXPECTED
;RECEIVED DATA
;UPDATA CHARACTER LENGTH
013222 005201 INC R1
013224 005204 INC R4
013226 006367 003130 ASL BYTCNT
013232 005300 DEC R0
013234 001325 BNE 1$
013236 104400 4$: SCOPE
000012 LINE=LINE+1
002000 BITX=BITX+BITX
SDATA3 \LINE,\BITX

;SINGLE LINE DATA TEST
;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 12
;LINE SPEED IS 9600 BAUD
;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
;TO 8 BITS
;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
;AT EACH CHARACTER LENGTH

013240 TS \XN,1,4$,1$
013240 012767 000340 164530 T53: MOV #340,PS ;DISABLE ALL INTERRUPTS
013246 012767 000001 003050 MOV #1,ICOUNT ;SET UP FOR 1 ITERATIONS
013254 012767 013456 003036 MOV #4$,ESCAPE ;SET UP TO ESCAPE TO NEXT TEST
;IF NB <1$>
013262 012767 013330 003032 MOV #1$,FREEZ1 ;SET UP TO LOOP WITH DATA ; 3
.ENDC
XN=XN+1
013270 000054 012777 004000 002760 MOV #BIT11,@DHSCR ;MASTER CLEAR INTERFACE
013276 005004 CLR R4 ;FIRST CHARACTER LENGTH CODE (5 BITS)
013300 012705 000012 MOV #12,R5 ;LINE 12 WILL BE TESTED
013304 012767 105000 003046 MOV #12*400+100000,RDATA ;SET EXPECTED LINE NUMBER
;AND VALID DATA FLAG
;EXPECTED DATA
;4 CHARACTER LENGTHS
;WILL BE TESTED
;FIRST CHARACTER LENGTH =5 BITS..
;LINE SPEED =9600 BAUD
;40 CHARACTERS AT 5 BITS
;SELECT LINE 12
013312 012700 000004 MOV #4,R0
013316 012701 033500 MOV #33500,R1
;SET LINE SPEED AND
;CHARACTER LENGTH
;ADDRESS OF TRANSMITTER
;DATA BUFFER
;400 (OCTAL) BYTES
;WILL BE TRANSMITTED
013322 012767 177740 003032 1$: MOV #-40,BYTCNT
013330 010577 002722 MOV R5,@DHSCR
013334 016702 003022 MOV BYTCNT,R2
013340 005402 NEG R2
013342 010177 002714 MOV R1,@DHLPR
013346 012777 016364 002710 MOV #TBUF,@DHBA
013354 016777 003002 002704 MOV BYTCNT,@DHBC

```

```

013362 012777 002000 002700      MOV    #2000,@DHBAR      ;START TRANSMITTER
013370 105777 002662      2$:   TSTB   @DHSCR          ;WAIT FOR DATA TO BE RECEIVED
013374 100375                BPL    2$
013376 017703 002656      MOV    @DHNRC,R3        ;GET RECEIVED DATA
013402 020367 002752      CMP    R3,RDATA        ;COMPER EXPECTED AND RECEIVED DATA
013406 001407                BEQ    3$
013410 005077 002654      CLR    @DHBAR          ;STOP TRANSMITTER
013414                HLT    2              ;DATA ERROR
013414 104002                EMT    2
013416 104410                SCOPE1
013420 012777 002000 002642      MOV    #2000,@DHBAR    ;CHECK FOR LOOP AT CURRENT SPEED
013426 105267 002726      3$:   INCB   RDATA        ;RESTART TRANSMITTER
013432 005302                DEC    R2              ;UPDATA EXPECTED DATA
013434 001355                BNE   2$
013436 105067 002716      CLRB  RDATA            ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
013442 005201                INC    R1              ;UPDATA CHARACTER LENGTH
013444 005204                INC    R4
013446 006367 002710      ASL   BYTCNT
013452 005300                DEC    R0
013454 001325                BNE   1$
013456 10440C      4$:   SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
013460 000013      SDATA3 \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 13
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

013460      TS \XN,1,4$,1$
013460 012767 000340 164310      T54:  MOV    #340,PS      ;DISABLE ALL INTERRUPTS
013466 012767 000001 002630      MOV    #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
013474 012767 013676 002616      MOV    #4$,ESCAPE    ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
013502 012767 013550 002612      MOV    #1$,FREEZ1    ;SET UP TO LOOP WITH DATA      ; 3
                                .ENDC
                                XN=XN+1
013510 012777 004000 002540      MOV    #BIT11,@DHSCR  ;MASTER CLEAR INTERFACE
013516 005004                CLR    R4              ;FIRST CHARACTER LENGTH CODE (5 BITS)
013520 012705 000013                MOV    #13,R5          ;LINE 13 WILL BE TESTED
013524 012767 105400 002626      MOV    #13*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS..
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
013532 012700 000004                MOV    #4,R0          ;SELECT LINE 13
013536 012701 033500                MOV    #33500,R1
013542 012767 177740 002612      MOV    #-40,BYTCNT
013550 010577 002502      1$:   MOV    R5,@DHSCR
013554 016702 002602      MOV    BYTCNT,R2
013560 005402                NEG    R2

```



```

013562 010177 002474          MOV    R1, @DHLPR          ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
013566 012777 016364 002470    MOV    #TBUF, @DHBA        ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
013574 016777 002562 002464    MOV    BYTCNT, @DHBC       ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
013602 012777 004000 002460    MOV    #4000, @DHBAR       ;START TRANSMITTER
013610 105777 002442          2$:  TSTB  @DHSCR           ;WAIT FOR DATA TO BE RECEIVED
013614 100375                    BPL    2$
013616 017707 002436          MOV    @DHNRC, R3          ;GET RECEIVED DATA
013622 020367 002532          CMP    R3, RDATA          ;COMPER EXPECTED AND RECEIVED DATA
013626 001407                    BEQ    3$
013630 005077 002434          CLR    @DHBAR            ;STOP TRANSMITTER
013634                    HLT    2
                                ;DATA ERROR
013634 104002                    EMT    2
013636 104410                    SCOPE1
013640 012777 004000 002422    MOV    #4000, @DHBAR       ;CHECK FOR LOOP AT CURRENT SPEED
013646 105267 002506          3$:  INCB  RDATA            ;RESTART TRANSMITTER
013652 005302                    DEC    R2
                                ;UPDATA EXPECTED DATA
013654 001355                    BNE   2$
013656 105067 002476          CLRB  RDATA
                                ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH
013662 005201                    INC    R1
013664 005204                    INC    R4
013666 006367 002470          ASL   BYTCNT
013672 005300                    DEC    R0
013674 001325                    BNE   1$
013676 104400          4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
013700 010000          SDATA3 \LINE, \BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 14
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

013700          TS \XN, 1, 4$, 1$
013700 012767 000340 164070    T55:  MOV    #340, PS        ;DISABLE ALL INTERRUPTS
013706 012767 000001 002410    MOV    #1, ICOUNT         ;SET UP FOR 1 ITERATIONS
013714 012767 014116 002376    MOV    #4$, ESCAPE        ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
013722 012767 013770 002372    MOV    #1$, FREEZ1        ;SET UP TO LOOP WITH DATA          ; 3
                                .ENDC
                                XN=XN+1
013730 012777 004000 002320    MOV    @BIT11, @DHSCR     ;MASTER CLEAR INTERFACE
013736 005004                    CLR    R4                 ;FIRST CHARACTER LENGTH CODE (5 BITS)
013740 012705 000014                    MOV    #14, R5            ;LINE 14 WILL BE TESTED
013744 012767 106000 002406    MOV    #14*400+100000, RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4CHARACTER LENGTHS
                                ;WILL BE TESTED

013752 012700 000004          MOV    #4, R0

```

```

013756 012701 033500      MOV      #33500,R1      ;FIRST CHARACTER LENGTH =5 BITS.,
                                ;LINE SPEED =9600 BAUD
013762 012767 177740 002372      MOV      #-40,BYTCNT    ;40 CHARACTERS AT 5 BITS
013770 010577 002262      1$:      MOV      R5,@DHSCR    ;SELECT LINE 14
013774 016702 002362      MOV      BYTCNT,R2
014000 005402      NEG      R2
014002 010177 002254      MOV      R1,@DHLPR      ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
014006 012777 016364 002250      MOV      #TBUF,@DHBA    ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
014014 016777 002342 002244      MOV      BYTCNT,@DHBC    ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
014022 012777 010000 002240      MOV      #10000,@DHBAR   ;START TRANSMITTER
014030 105777 002222      2$:      TSTB     @DHSCR        ;WAIT FOR DATA TO BE RECEIVED
014034 100375      BPL      2$
014036 017703 002216      MOV      @DHNRC,R3      ;GET RECEIVED DATA
014042 020367 002312      CMP      R3,RDATA       ;COMPER EXPECTED AND RECEIVED DATA
014046 001407      BEQ      3$
014050 005077 002214      CLR      @DHBAR        ;STOP TRANSMITTER
014054      HLT      2        ;DATA ERROR
014054 104002      EMT      2
014056 104410      SCOPE1
014060 012777 010000 002202      MOV      #10000,@DHBAR   ;CHECK FOR LOOP AT CURRENT SPEED
014066 105267 002266      3$:      INCB     RDATA         ;RESTART TRANSMITTER
014072 005302      DEC      R2            ;UPDATA EXPECTED DATA
014074 001355      BNE      2$
014076 105067 002256      CLRB     RDATA         ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
014102 005201      INC      R1            ;UPDATA CHARACTER LENGTH
014104 005204      INC      R4
014106 006367 002250      ASL      BYTCNT
014112 005300      DEC      R0
014114 001325      BNE      1$
014116 104400      4$:      SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
014120      SDATA3 \LINE,\BITX

                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 15
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

014120      TS \XN,1,4$,1$
014120 012767 000340 163650      T56:     MOV      #340,PS      ;DISABLE ALL INTERRUPTS
014126 012767 000001 002170      MOV      #1,ICOUNT      ;SET UP FOR 1 ITERATIONS
014134 012767 014336 002156      MOV      #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>
014142 012767 014210 002152      MOV      #1$,FREEZ1     ;SET UP TO LOOP WITH DATA ; 5
                                .ENDC
                                XN=XN+1
014150 012777 004000 002100      MOV      #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
014156 005004      CLR      R4            ;FIRST CHARACTER LENGTH CODE (5 BITS)
014160 012705 000015      MOV      #15,R5        ;LINE 15 WILL BE TESTED

```

```

014164 012767 106400 002166      MOV      #15*400+100000,RDATA      ;SET EXPECTED LINE NUMBER
                                           ;AND VALID DATA FLAG
014172 012700 000004      MOV      #4,R0                    ;EXPECTED DATA
014176 012701 033500      MOV      #33500,R1                ;4CHARACTER LENGTHS
                                           ;WILL BE TESTED
014202 012767 177740 002152      MOV      #40,BYTCNT               ;FIRST CHARACTER LENGTH =5 BITS..
014210 010577 002042      1$:   MOV      R5,@DHSCR           ;LINE SPEED =9600 BAUD
014214 016702 002142      MOV      BYTCNT,R2                ;40 CHARACTERS AT 5 BITS
014220 005402      NEG      R2                        ;SELECT LINE 15
014222 010177 002034      MOV      R1,@DHLPR                ;SET LINE SPEED AND
                                           ;CHARACTER LENGTH
014226 012777 016364 002030      MOV      #TBUF,@DHBA              ;ADDRESS OF TRANSMITTER
014234 016777 002122 002024      MOV      BYTCNT,@DHBC             ;DATA BUFFER
                                           ;400 (OCTAL) BYTES
014242 012777 020000 002020      MOV      #20000,@DHBAR            ;WILL BE TRANSMITTED
014250 105777 002002      2$:   TSTB      @DHSCR              ;START TRANSMITTER
014254 100375      RPL      2$                        ;WAIT FOR DATA TO BE RECEIVED
014256 017703 001776      MOV      @DHNR,R3                 ;GET RECEIVED DATA
014262 020367 002072      CMP      R3,RDATA                 ;COMPER EXPECTED AND RECEIVED DATA
014266 001407      BEQ      3$                        ;STOP TRANSMITTER
014270 005077 001774      CLR      @DHBAR                    ;DATA ERROR
014274      HLT      2
014274 104002      EMT      2
014276 104410      SCOPE1
014300 012777 020000 001762      MOV      #20000,@DHBAR            ;CHECK FOR LOOP AT CURRENT SPEED
014306 105267 002046      3$:   INCB      RDATA                ;RESTART TRANSMITTER
014312 005302      DEC      R2                        ;UPDATA EXPECTED DATA
014314 001355      BNE      2$
014316 105067 002036      CLRB      RDATA                    ;INITIALIZE EXPECTED
                                           ;RECEIVED DATA
014322 005201      INC      R1                        ;UPDATA CHARACTER LENGTH
014324 005204      INC      R4
014326 006367 002030      ASL      BYTCNT
014332 005300      DEC      R0
014334 001325      BNE      1$
014336 104400      4$:   SCOPE
014340 000016      LINE=LINE+1
014340 040000      BITX=BITX+BITX
                                SDATA3 \LINE,\BITX
                                ;SINGLE LINE DATA TEST
                                ;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 16
                                ;LINE SPEED IS 9600 BAUD
                                ;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
                                ;TO 8 BITS
                                ;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
                                ;AT EACH CHARACTER LENGTH

014340      TS \XN,1,4$,1$
014340 012767 000340 163430      T57:   MOV      #340,PS              ;DISABLE ALL INTERRUPTS
014346 012767 000001 001750      MOV      #1,ICOUNT                ;SET UP FOR 1 ITERATIONS
014354 012767 014556 001736      MOV      #4$,ESCAPE                ;SET UP TO ESCAPE TO NEXT TEST
                                .IF NB <1$>

```

```

014362 012767 014430 001732      MOV      #1$,FREEZ1      ;SET UP TO LOOP WITH DATA      ; 3
      .ENOC
      XN=XN+1
014370 012777 004000 001660      MOV      @BIT11,@DHSCR      ;MASTER CLEAR INTERFACE
014376 005004      CLR      R4      ;FIRST CHARACTER LENGTH CODE (5 BITS)
014400 0127C5 000016      MOV      #16,R5      ;LINE 16 WILL BE TESTED
014404 012767 107000 001746      MOV      #16*400*100000,RDATA
      ;SET EXPECTED LINE NUMBER
      ;AND VALID DATA FLAG
      ;EXPECTED DATA
      ;4CHARACTER LENGTHS
      ;WILL BE TESTED
014412 012700 000004      MOV      #4,R0
014416 012701 033500      MOV      #33500,R1
      ;FIRST CHARACTER LENGTH =5 BITS,,
      ;LINE SPEED =9600 BAUD
014422 012767 177740 001732      MOV      # 40,BYTCNT      ;40 CHARACTERS AT 5 BITS
014430 010577 001622      1$: MOV      R5,@DHSCR      ;SELECT LINE 16
014434 016702 001722      MOV      BYTCNT,R2
014440 005402      NEG      R2
014442 010177 001614      MOV      R1,@DHLPR
      ;SET LINE SPEED AND
      ;CHARACTER LENGTH
      ;ADDRESS OF TRANSMITTER
      ;DATA BUFFER
014446 012777 016364 001610      MOV      @TBUF,@DHBA
      ;400 (OCTAL) BYTES
      ;WILL BE TRANSMITTED
014462 012777 040000 001600      MOV      #40000,@DHBAR
      ;START TRANSMITTER
014470 105777 001562      2$: TSTB   @DHSCR      ;WAIT FOR DATA TO BE RECEIVED
014474 100375      BPL      2$
014476 017703 001556      MOV      @DHNRC,R3      ;GET RECEIVED DATA
014502 020367 001652      CMP      R3,RDATA      ;COMPER EXPECTED AND RECEIVED DATA
014506 001407      BEQ      3$
014510 005077 001554      CLR      @DHBAR
      ;STOP TRANSMITTER
014514      HLT      2      ;DATA ERROR
014514 104002      EMT      2
014516 104410      SCOPE1
014520 012777 040000 001542      MOV      #40000,@DHBAR
      ;CHECK FOR LOOP AT CURRENT SPEED
014526 105267 001626      3$: INCB   RDATA      ;RESTART TRANSMITTER
014532 005302      DEC      R2      ;UPDATA EXPECTED DATA
014534 001355      BNE      2$
014536 105067 001616      CLRB   RDATA
      ;INITIALIZE EXPECTED
      ;RECEIVED DATA
      ;UPDATA CHARACTER LENGTH
014542 005201      INC      R1
014544 005204      INC      R4
014546 006367 001610      ASL     BYTCNT
014552 005300      DEC     R0
014554 001325      BNE     1$
014556 104400      4$: SCOPE
      LINE=LINE+1
      BITX=BITX+BITX
      SDATA3 \LINE,\BITX
014560 100000

```

```

;SINGLE LINE DATA TEST
;TRANSMIT A BLOCK OF 400 (OCTAL) CHARACTERS ON LINE 17
;LINE SPEED IS 9600 BAUD
;CHARACTER LENGTH WILL START AT 5 BITS AND BE INCREMENTED
;TO 8 BITS
;A BLOCK OF 400 CHARACTERS WILL BE TRANSMITTED
;AT EACH CHARACTER LENGTH

```

```

014560          TS \XN,1,4$,1$
014560 012767 000340 163210 T60:  MOV    #340,PS          ;DISABLE ALL INTERRUPTS
014566 012767 000001 00153C      MOV    #1,ICOUNT       ;SET UP FOR 1 ITERATIONS
014574 012767 014776 001516      MOV    #4$,ESCAPE     ;SET UP TO ESCAPE TO NEXT TEST

014602 012767 014650 001512      MOV    #1$,FREEZ1     ;SET UP TO LOOP WITH DATA      ; 3
                                .IF NB <1$>
                                .ENDC
                                XN=XN+1
014610 000061          MOV    #9IT11,@DHSCR   ;MASTER CLEAR INTERFACE
014616 012777 004000 001440      CLR    F4             ;FIRST CHARACTER LENGTH CODE (5 BITS)
014620 012705 000017          MOV    #17,R5         ;LINE 17 WILL BE TESTED
014624 012767 107400 001526      MOV    #17*400+100000,RDATA
                                ;SET EXPECTED LINE NUMBER
                                ;AND VALID DATA FLAG
                                ;EXPECTED DATA
                                ;4 CHARACTER LENGTHS
                                ;WILL BE TESTED
                                ;FIRST CHARACTER LENGTH =5 BITS,,
                                ;LINE SPEED =9600 BAUD
                                ;40 CHARACTERS AT 5 BITS
                                ;SELECT LINE 17

014632 012700 000004          MOV    #4,R0
                                ;SET LINE SPEED AND
                                ;CHARACTER LENGTH
                                ;ADDRESS OF TRANSMITTER
                                ;DATA BUFFER
                                ;400 (OCTAL) BYTES
                                ;WILL BE TRANSMITTED
                                ;START TRANSMITTER
                                ;WAIT FOR DATA TO BE RECEIVED

014636 012701 033500          MOV    #33500,R1
                                ;GET RECEIVED DATA
                                ;COMPER EXPECTED AND RECEIVED DATA

014642 012767 177740 001512      MOV    #-40,BYTCNT
014650 010577 001402          1$:  MOV    R5,@DHSCR
014654 016702 001502          MOV    BYTCNT,R2
014660 005402          NEG    R2
014662 010177 001374          MOV    R1,@DHLPR
                                ;STOP TRANSMITTER
                                ;DATA ERROR

014666 012777 016364 001370      MOV    #TBUF,@DHBA
                                ;CHECK FOR LOOP AT CURRENT SPEED
                                ;RESTART TRANSMITTER
                                ;UPDATA EXPECTED DATA

014674 016777 001462 001364      MOV    BYTCNT,@DHBC
                                ;INITIALIZE EXPECTED
                                ;RECEIVED DATA
                                ;UPDATA CHARACTER LENGTH

014702 012777 100000 001360      MOV    #100000,@DHBAR
014710 105777 001342          2$:  TSTB  @DHSCR
014714 100375          BPL    2$
014716 017703 001336          MOV    @DHNRC,R3
014722 020367 001432          CMP    R3,RDATA
014726 001407          BEQ    3$
014730 005077 001334          CLR    @DHBAR
014734          HLT    2
014734 104002          EMT    2
014736 104410          SCOPE1
014740 012777 100000 001322      MOV    #100000,@DHBAR
014746 105267 001406          3$:  INCB  RDATA
014752 005302          DEC    R2
014754 001355          BNE    2$
014756 105067 001376          CLRB  RDATA

014762 005201          INC    R1
014764 005204          INC    R4
014766 006367 001370          ASL   BYTCNT
014772 005300          DEC    R0
014774 001325          BNE    1$
014776 104400          4$:  SCOPE
                                LINE=LINE+1
                                BITX=BITX+BITX
                                XDATA=0
000020
000000
000000
43      000000

```

```

1
2 015000      .EOP      †/BEGIN/

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

015000  104401      EOP:      TYPE          ;TYPE NAME OF TEST
015002  017347      MEPASS
015004  005067  001344  CLR          LAST          ;CLEAR LAST ERROR PC
015010  005067  001274  CLR          ERRFLG       ;CLEAR ERROR FLAG
015014  005267  001272  INC          PASCNT       ;UPDATE PASS COUNT
015020  005767  163756  TST          LIGHTS      ; ARE WE USING LIGHTS?      ; 4
015024  001005      BNE          2$          ; BRANCH IF WE ARE          ; 6
015026  104401      TYPE
015030  017362      PASTXT
015032  104402      OCTASC
015034  015072      PASARG
015036  000403      BR          3$          ; TYPE PASCOUNT MESSAGE    ; 5
015040      2$:
015040  016767  001246  163734  MOV          PASCNT,LIGHTS ; PRINT PASCOUNT            ; 4
015046      3$:
015046  013701  000042  MOV          @#42,R1      ; CONTINUE                  ; 4
015052  001405      BEQ          RESTRT      ;DISPLAY PASS COUNT        ; 4
015054  000005      RESET
015056  004711      LOGICAL: JSR          PC,(R1) ;CHECK FOR ACT-11 OR DDP    ; 4
015060  000240      NOP
015062  000240      NOP
015064  000240      NOP
015066  000167  164214  RESTRT: JMP          BEGIN
015072  000001      PASARG: .WORD        1          ; PARAMETERS TO PRINT PASCOUNT ; 5
015074  006          002      .BYTE        6,2
015076  016312      .WORD        PASCNT      ; 5
3 015100      .SCOPE

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

015100  032777  002000  163672  SCOPER: BIT          @SW10,@SWR      ; 4
015106  001030      BNE          4$
015110  032777  040000  163662  1$: BIT          @SW14,@SWR      ; 4
015116  001021      BNE          3$
015120  032777  004000  163652  BIT          @SW11,@SWR      ; 4
015126  001006      BNE          2$
015130  005267  001172  INC          LPCNT
015134  026767  001166  001162  CMP          LPCNT,ICOUNT
015142  001007      BNE          3$
015144  005067  001156      2$: CLR          LPCNT
015150  005067  001134  CLR          ERRFLG
015154  011667  001136  MOV          (SP),RETRN
015160  000002      RTI
015162  016716  001130      3$: MOV          RETRN,(SP)
015166  000002      RTI
015170  005767  001114      4$: TST          ERRFLG
015174  001745      BEQ          1$

```

015176 000762
4 015200

BR 2\$
.SCOP1

;CHECK FOR FREEZE ON CURRENT DATA

015200 032777 001000 163572 SCOP1R: BIT #SW09,@SWR
015206 001402 BEQ 1\$
015210 016716 001106 MOV FREEZ1,(SP)
015214 000002 1\$: RTI

; 4

1 015216

.ERROR

;ERROR HANDLER

```

015216 032777 020000 163554 ERRORS: BIT #SW13,@SWR ; 4
015224 001055 BNE HALTS
015226 021667 001122 CMP (SP),LAST
015232 001404 BEQ 1$
015234 011667 001114 MOV (SP),LAST
015240 005067 001044 CLR ERRFLG
015244 104406 1$: SAV05P
015246 011605 MOV (SP),R5
015250 162705 000002 SUB #2,R5
015254 011504 MOV (R5),R4
015256 006304 ASL R4
015260 006304 ASL R4
015262 042704 177001 BIC #177001,R4
015266 062704 017502 ADD #ERRTAB,R4
015272 012467 000040 MOV (R4)+,ERRMSG
015276 011467 000052 MOV (R4),DATABP
015302 005767 001002 TST ERRFLG
015306 001403 BEQ TYPMSG
015310 005767 000040 TST DATABP
015314 001011 BNE TYPDAT
015316 104401 TYPMSG: TYPE ; 3
015320 017257 MCRLF ; 5
015322 104402 OCTASC ; 5
015324 015422 ERTABO ; 5
015326 012767 000001 000754 MOV #1,ERRFLG
015334 104401 TYPE
015336 000000 ERRMSG: 0
015340 005767 000010 TYPDAT: TST DATABP
015344 001404 BEQ RESREG
015346 104401 TYPE ; 5
015350 017257 MCRLF ; 5
015352 104402 OCTASC
015354 000000 DATABP: 0
015356 104407 RESREG: RES05
015360 005777 163414 HALTS: TST @SWR ; 4
015364 100005 BPL EXITER
015366 010046 PUSHRO
015370 016600 000002 MOV 2(SP),R0
015374 000000 HALT
015376 012600 POPRO
015400 005267 000710 EXITER: INC ERRCNT
015404 032777 002000 163366 BIT #SW10,@SWR ; 4
015412 001402 BEQ 1$
015414 016716 000700 MOV ESCAPE,(SP)
015420 000002 1$: RTI
015422 000001 ERTABO: 1
015424 006 002 .BYTE 6,2
015426 016346 SAVPC

```


015430

.TRPSRV

```

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

```

; 3

```

015430 011646
015432 162716 000002
015436 017616 000000
015442 006316
015444 042716 177001
015450 062716 017422
015454 017616 000000
015460 000136

```

```

RPSRV: MOV (SP),-(SP) ;GET PC OF RETURN
SUB #2,(SP) ;=PC OF TRAP
MOV @17616,(SP) ;GET TRP
TRPOK: ASL (SP) ;MULTIPLY TRAP ARG BY 2
BIC #177001,(SP) ;CLEAR UNWANTED BITS
ADD #TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
MOV @17616,(SP) ;SUBROUTINE ADDRESS
JMP @17616 ;GO TO SUBROUTINE

```

2 015462

.SAVREG

;SAVE PC OF TEST THAT FAILED AND R0-R5

```

015462 016667 000004 000656 SV05P: MOV 4(SP),SAVPC

```

;SAVE R0-R5

```

015470 010567 000646 SV05: MOV R5,SAVR5
015474 010467 000640 MOV R4,SAVR4
015500 010367 000632 MOV R3,SAVR3
015504 010267 000624 MOV R2,SAVR2
015510 010167 000616 MOV R1,SAVR1
015514 010067 000610 MOV R0,SAVR0
015520 000002 RTI

```

; 3

3 015522

.RESREG

;RESTORE R0-R5

```

015522 016700 000602 RS05: MOV SAVR0,R0
015526 016701 000600 MOV SAVR1,R1
015532 016702 000576 MOV SAVR2,R2
015536 016703 000574 MOV SAVR3,R3
015542 016704 000572 MOV SAVR4,R4
015546 016705 000570 MOV SAVR5,R5
015552 000002 RTI

```

1 015554

.TYPER

;TELETYPE OUTPUT ROUTINE

015554 017605 000000
 015560 062716 000002
 015564 105777 000462
 015570 100375
 015572 105715
 015574 001001
 015576 000002
 015600 112577 000450
 015604 000767
 2 015606

TYPER: MOV @ (SP),R5
 ADD #2,(SP)
 1\$: TSTB @TPCSR
 BPL 1\$
 TSTB (R5)
 BNE 2\$
 RTI
 2\$: MOVB (R5)+,@TPDBR
 BR 1\$

; 3

.INSTRG

;ASCII SIPPING INPUT ROUTINE

015606 017667 000000 000006
 015614 062716 000002
 015620 104401
 015622 000000
 015624 012704 017444
 015630 012703 000007
 015634 105777 000406
 015640 100375
 015642 117714 000402
 015646 142714 000200
 015652 122427 000015
 015656 001413
 015660 117777 000364 000366
 015666 105777 000360
 015672 100375
 015674 005303
 015676 001356
 015700 104401
 015702 017253
 015704 000745
 015706 000002

INSTRG: MOV @ (SP),MSG
 ADD #2,(SP)
 INSTR1: TYPE
 MSG: 0
 MOV #INBUF,R4
 MOV #7,R3
 1\$: TSTB @TKCSR
 BPL 1\$
 MOVB @TKDBR,(R4)
 BICB #200,(R4)
 CMPB (R4)+,#15
 BEQ INSTR2
 MOVB @TKDBR,@TPDBR
 2\$: TSTB @TPCSR
 BPL 2\$
 DEC R3
 BNE 1\$
 INSTR2: TYPE
 MQM
 BR INSTR1
 INSTR2: RTI

1 015710

.PARAMS

;CONVERT ASCII STRING TO OCTAL

; 3

015710 011605
 015712 012567 000146
 015716 012567 000144
 015722 012567 000142
 015726 112567 000140
 015732 112567 000135
 015736 010516
 015740 005005
 015742 012704 017444
 015746 122714 000015
 015752 001420
 015754 121427 000060
 015760 002415
 015762 121427 000067
 015766 003012
 015770 142714 000060
 015774 152405
 015776 122714 000015
 016002 001406
 016004 006305
 016006 006305
 016010 006305
 016012 000760
 016014 104404
 016016 000750

PARAMS: MOV (SP),R5
 MOV (R5)+,LOLIM
 MOV (R5)+,HILIM
 MOV (R5)+,DEVADR
 MOVB (R5)+,LOBITS
 MOVB (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

016020 020567 000042
 016024 101373
 016026 020567 000032
 016032 103770
 016034 136705 000032
 016040 001365

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

; 3

;STORE NUMBER AT SPECIFIED ADDRESS

016042 016704 000022
 016046 010524
 016050 062705 000002
 016054 105367 000013
 016060 001372
 016062 000002
 016064 000000
 016066 000000
 016070 000000
 016072 000000
 016073

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

016074

.OCTASC

:CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

016074	017601	000000	OCTASN:	MOV	@(SP),R1	
016100	062716	000002		ADD	#2,(SP)	; 5
016104	012167	000130		MOV	(R1)+,WRDCNT	
016110	112167	000126	1\$:	MOVB	(R1)+,CHRCNT	
016114	112167	000123		MOVB	(R1)+,SPACNT	
016120	013167	000120		MOV	@(R1)+,BINWRD	; 3
016124	016704	000114	2\$:	MOV	BINWRD,R4	
016130	116705	000106		MOVB	CHRCNT,R5	
016134	012700	017456		MOV	#TEMP,R0	
016140	010403		3\$:	MOV	R4,R3	
016142	042703	177770		BIC	#177770,R3	
016146	062703	000260		ADD	#260,R3	
016152	110320			MOVB	R3,(R0)+	
016154	006204			ASR	R4	
016156	006204			ASR	R4	
016160	006204			ASR	R4	
016162	005305			DEC	R5	
016164	001365			BNE	3\$	
016166	012703	017470		MOV	#MDATA,R3	
016172	114023		4\$:	MOVB	-(R0),(R3)+	
016174	105367	000042		DECB	CHRCNT	
016200	001374			BNE	4\$	
016202	105767	000035		TSTB	SPACNT	
016206	001405			BEQ	6\$	
016210	112723	000240	5\$:	MOVB	#240,(R3)+	
016214	105367	000023		DECB	SPACNT	
016220	001373			BNE	5\$	
016222	105013		6\$:	CLRB	(R3)	
016224	104401			TYPE		
016226	017470			MDATA		
016230	005367	000004		DEC	WRDCNT	
016234	001325			BNE	1\$	
016236	000002			RTI		
016240	000000		WRDCNT:	0		
016242	000000		CHRCNT:	0		
	016243		SPACNT=	CHRCNT+1		
016244	000000		BINWRD:	0		

```

016246          .POINT  + 'DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSP,DHSLR,DHRVEC,DHRLVL,DHTVEC,DHTLVL/
                  ;INDIRECT POINTERS
                  ; 3

016246 177560    TKCSR:  177560
016250 177562    TKDBR:  177562
016252 177564    TPCSR:  177564
016254 177566    TPDBR:  177566
TLVL>          .IRP    A      <DHSCR,DHNRC,DHLPR,DHBA,DHBC,DHBAR,DHBCR,DHSSR,DHSLR,DHRVEC,DHRLVL,DHTVEC,DH
                  A:      0
                  .ENDM
016256 000000    DHSCR:  0
016260 000000    DHNRC:  0
016262 000000    DHLPR:  0
016264 000000    DHBA:   0
016266 000000    DHBC:   0
016270 000000    DHBAR:  0
016272 000000    DHBCR:  0
016274 000000    DHSSR:  0
016276 000000    DHSLR:  0
016300 000000    DHRVEC: 0
016302 000000    DHRLVL: 0
016304 000000    DHTVEC: 0
016306 000000    DHTLVL: 0
2 016310          .VARIA  + /TDATA,RDATA,BYTCNT/
                  ;PROGRAM VARIABLES

016310 000000    ERRFLG: 0          ;ERROR FLAG
016312 000000    PASCNT: 0          ;PASS COUNT
016314 000000    ERRCNT: 0          ;ERROR COUNT
016316 000000    RETRN:  0          ;SCOPE RETURN ADDRESS FOR TEST LOOPING
016320 000000    ESCAPE: 0          ;ADDRESS FOR ERROR ESCAPE
016322 000000    FREEZ1: 0          ;DATA LOOPING RETURN ADDRESS
016324 000000    ICOUNT: 0          ;ITERATION COUNT FOR TEST IN PROGRESS
016326 000000    LPCNT:  0          ;NUMBER OF ITERATIONS THIS TEST
016330 000000    SAVR0:  0          ;R0 SAVE AREA
016332 000000    SAVR1:  0          ;R1 SAVE AREA
016334 000000    SAVR2:  0          ;R2 SAVE AREA
016336 000000    SAVR3:  0          ;R3 SAVE ARE
016340 000000    SAVR4:  0          ;R4 SAVE AREA
016342 000000    SAVR5:  0          ;R5 SAVE AREA
016344 000000    SAVSP:  0          ;STACK POINTER SAVE AREA
016346 000000    SAVPC:  0          ;CALLING ROUTINE SAVE AREA
016350 000000    INIFLG: 0          ;PROGRAM INITIALIZATION FLAG
016352 000000    STFLG:  0          ;PROGRAM START FLAG
016354 000000    LAST:   0          ;LAST ERROR PC
                  .IRP    A      <TDATA,RDATA,BYTCNT>
                  A:      0
                  .ENDM
016356 000000    TDATA:  0
016360 000000    RDATA:  0
016362 000000    BYTCNT: 0

```

: 3

```
1 016364 000400      TBUF:  .REPT  400
2                .BYTE  XDATA
3                .NLIST
4                XDATA=XDATA+1
5                .LIST
6                .ENDR
016364      000      .BYTE  XDATA
                000001 XDATA=XDATA+1
016365      001      .BYTE  XDATA
                000002 XDATA=XDATA+1
016366      002      .BYTE  XDATA
                000003 XDATA=XDATA+1
016367      003      .BYTE  XDATA
                000004 XDATA=XDATA+1
016370      004      .BYTE  XDATA
                000005 XDATA=XDATA+1
016371      005      .BYTE  XDATA
                000006 XDATA=XDATA+1
016372      006      .BYTE  XDATA
                000007 XDATA=XDATA+1
016373      007      .BYTE  XDATA
                000010 XDATA=XDATA+1
016374      010      .BYTE  XDATA
                000011 XDATA=XDATA+1
016375      011      .BYTE  XDATA
                000012 XDATA=XDATA+1
016376      012      .BYTE  XDATA
                000013 XDATA=XDATA+1
016377      013      .BYTE  XDATA
                000014 XDATA=XDATA+1
016400      014      .BYTE  XDATA
                000015 XDATA=XDATA+1
016401      015      .BYTE  XDATA
                000016 XDATA=XDATA+1
016402      016      .BYTE  XDATA
                000017 XDATA=XDATA+1
016403      017      .BYTE  XDATA
                000020 XDATA=XDATA+1
016404      020      .BYTE  XDATA
                000021 XDATA=XDATA+1
016405      021      .BYTE  XDATA
                000022 XDATA=XDATA+1
016406      022      .BYTE  XDATA
                000023 XDATA=XDATA+1
016407      023      .BYTE  XDATA
                000024 XDATA=XDATA+1
016410      024      .BYTE  XDATA
                000025 XDATA=XDATA+1
016411      025      .BYTE  XDATA
                000026 XDATA=XDATA+1
016412      026      .BYTE  XDATA
                000027 XDATA=XDATA+1
016413      027      .BYTE  XDATA
                000030 XDATA=XDATA+1
016414      030      .BYTE  XDATA
                000031 XDATA=XDATA+1
016415      031      .BYTE  XDATA
```

016416	000032	XDATA=XDATA+1
	032	.BYTE XDATA
016417	000033	XDATA=XDATA+1
	033	.BYTE XDATA
016420	000034	XDATA=XDATA+1
	034	.BYTE XDATA
016421	000035	XDATA=XDATA+1
	035	.BYTE XDATA
016422	000036	XDATA=XDATA+1
	036	.BYTE XDATA
016423	000037	XDATA=XDATA+1
	037	.BYTE XDATA
016424	000040	XDATA=XDATA+1
	040	.BYTE XDATA
016425	000041	XDATA=XDATA+1
	041	.BYTE XDATA
016426	000042	XDATA=XDATA+1
	042	.BYTE XDATA
016427	000043	XDATA=XDATA+1
	043	.BYTE XDATA
016430	000044	XDATA=XDATA+1
	044	.BYTE XDATA
016431	000045	XDATA=XDATA+1
	045	.BYTE XDATA
016432	000046	XDATA=XDATA+1
	046	.BYTE XDATA
016433	000047	XDATA=XDATA+1
	047	.BYTE XDATA
016434	000050	XDATA=XDATA+1
	050	.BYTE XDATA
016435	000051	XDATA=XDATA+1
	051	.BYTE XDATA
016436	000052	XDATA=XDATA+1
	052	.BYTE XDATA
016437	000053	XDATA=XDATA+1
	053	.BYTE XDATA
016440	000054	XDATA=XDATA+1
	054	.BYTE XDATA
016441	000055	XDATA=XDATA+1
	055	.BYTE XDATA
016442	000056	XDATA=XDATA+1
	056	.BYTE XDATA
016443	000057	XDATA=XDATA+1
	057	.BYTE XDATA
016444	000060	XDATA=XDATA+1
	060	.BYTE XDATA
016445	000061	XDATA=XDATA+1
	061	.BYTE XDATA
016446	000062	XDATA=XDATA+1
	062	.BYTE XDATA
016447	000063	XDATA=XDATA+1
	063	.BYTE XDATA
016450	000064	XDATA=XDATA+1
	064	.BYTE XDATA
016451	000065	XDATA=XDATA+1
	065	.BYTE XDATA
	000066	XDATA=XDATA+1

016452	066	.BYTE XDATA
	000067	XDATA=XDATA+1
016453	067	.BYTE XDATA
	000070	XDATA=XDATA+1
016454	070	.BYTE XDATA
	000071	XDATA=XDATA+1
016455	071	.BYTE XDATA
	000072	XDATA=XDATA+1
016456	072	.BYTE XDATA
	000073	XDATA=XDATA+1
016457	073	.BYTE XDATA
	000074	XDATA=XDATA+1
016460	074	.BYTE XDATA
	000075	XDATA=XDATA+1
016461	075	.BYTE XDATA
	000076	XDATA=XDATA+1
016462	076	.BYTE XDATA
	000077	XDATA=XDATA+1
016463	077	.BYTE XDATA
	000100	XDATA=XDATA+1
016464	100	.BYTE XDATA
	000101	XDATA=XDATA+1
016465	101	.BYTE XDATA
	000102	XDATA=XDATA+1
016466	102	.BYTE XDATA
	000103	XDATA=XDATA+1
016467	103	.BYTE XDATA
	000104	XDATA=XDATA+1
016470	104	.BYTE XDATA
	000105	XDATA=XDATA+1
016471	105	.BYTE XDATA
	000106	XDATA=XDATA+1
016472	106	.BYTE XDATA
	000107	XDATA=XDATA+1
016473	107	.BYTE XDATA
	000110	XDATA=XDATA+1
016474	110	.BYTE XDATA
	000111	XDATA=XDATA+1
016475	111	.BYTE XDATA
	000112	XDATA=XDATA+1
016476	112	.BYTE XDATA
	000113	XDATA=XDATA+1
016477	113	.BYTE XDATA
	000114	XDATA=XDATA+1
016500	114	.BYTE XDATA
	000115	XDATA=XDATA+1
016501	115	.BYTE XDATA
	000116	XDATA=XDATA+1
016502	116	.BYTE XDATA
	000117	XDATA=XDATA+1
016503	117	.BYTE XDATA
	000120	XDATA=XDATA+1
016504	120	.BYTE XDATA
	000121	XDATA=XDATA+1
016505	121	.BYTE XDATA
	000122	XDATA=XDATA+1
016506	122	.BYTE XDATA

016507	000123	XDATA=XDATA+1
	123	.BYTE XDATA
016510	000124	XDATA=XDATA+1
	124	.BYTE XDATA
016511	000125	XDATA=XDATA+1
	125	.BYTE XDATA
016512	000126	XDATA=XDATA+1
	126	.BYTE XDATA
016513	000127	XDATA=XDATA+1
	127	.BYTE XDATA
016514	000130	XDATA=XDATA+1
	130	.BYTE XDATA
016515	000131	XDATA=XDATA+1
	131	.BYTE XDATA
016516	000132	XDATA=XDATA+1
	132	.BYTE XDATA
016517	000133	XDATA=XDATA+1
	133	.BYTE XDATA
016520	000134	XDATA=XDATA+1
	134	.BYTE XDATA
016521	000135	XDATA=XDATA+1
	135	.BYTE XDATA
016522	000136	XDATA=XDATA+1
	136	.BYTE XDATA
016523	000137	XDATA=XDATA+1
	137	.BYTE XDATA
016524	000140	XDATA=XDATA+1
	140	.BYTE XDATA
016525	000141	XDATA=XDATA+1
	141	.BYTE XDATA
016526	000142	XDATA=XDATA+1
	142	.BYTE XDATA
016527	000143	XDATA=XDATA+1
	143	.BYTE XDATA
016530	000144	XDATA=XDATA+1
	144	.BYTE XDATA
016531	000145	XDATA=XDATA+1
	145	.BYTE XDATA
016532	000146	XDATA=XDATA+1
	146	.BYTE XDATA
016533	000147	XDATA=XDATA+1
	147	.BYTE XDATA
016534	000150	XDATA=XDATA+1
	150	.BYTE XDATA
016535	000151	XDATA=XDATA+1
	151	.BYTE XDATA
016536	000152	XDATA=XDATA+1
	152	.BYTE XDATA
016537	000153	XDATA=XDATA+1
	153	.BYTE XDATA
016540	000154	XDATA=XDATA+1
	154	.BYTE XDATA
016541	000155	XDATA=XDATA+1
	155	.BYTE XDATA
016542	000156	XDATA=XDATA+1
	156	.BYTE XDATA
	000157	XDATA=XDATA+1

016543	157	.BYTE XDATA
	000160	XDATA=XDATA+1
016544	160	.BYTE XDATA
	000161	XDATA=XDATA+1
016545	161	.BYTE XDATA
	000162	XDATA=XDATA+1
016546	162	.BYTE XDATA
	000163	XDATA=XDATA+1
016547	163	.BYTE XDATA
	000164	XDATA=XDATA+1
016550	164	.BYTE XDATA
	000165	XDATA=XDATA+1
016551	165	.BYTE XDATA
	000166	XDATA=XDATA+1
016552	166	.BYTE XDATA
	000167	XDATA=XDATA+1
016553	167	.BYTE XDATA
	000170	XDATA=XDATA+1
016554	170	.BYTE XDATA
	000171	XDATA=XDATA+1
016555	171	.BYTE XDATA
	000172	XDATA=XDATA+1
016556	172	.BYTE XDATA
	000173	XDATA=XDATA+1
016557	173	.BYTE XDATA
	000174	XDATA=XDATA+1
016560	174	.BYTE XDATA
	000175	XDATA=XDATA+1
016561	175	.BYTE XDATA
	000176	XDATA=XDATA+1
016562	176	.BYTE XDATA
	000177	XDATA=XDATA+1
016563	177	.BYTE XDATA
	000200	XDATA=XDATA+1
016564	200	.BYTE XDATA
	000201	XDATA=XDATA+1
016565	201	.BYTE XDATA
	000202	XDATA=XDATA+1
016566	202	.BYTE XDATA
	000203	XDATA=XDATA+1
016567	203	.BYTE XDATA
	000204	XDATA=XDATA+1
016570	204	.BYTE XDATA
	000205	XDATA=XDATA+1
016571	205	.BYTE XDATA
	000206	XDATA=XDATA+1
016572	206	.BYTE XDATA
	000207	XDATA=XDATA+1
016573	207	.BYTE XDATA
	000210	XDATA=XDATA+1
016574	210	.BYTE XDATA
	000211	XDATA=XDATA+1
016575	211	.BYTE XDATA
	000212	XDATA=XDATA+1
016576	212	.BYTE XDATA
	000213	XDATA=XDATA+1
016577	213	.BYTE XDATA

016600	000214	XDATA=XDATA+1
	214	.BYTE XDATA
016601	000215	XDATA=XDATA+1
	215	.BYTE XDATA
016602	000216	XDATA=XDATA+1
	216	.BYTE XDATA
016603	000217	XDATA=XDATA+1
	217	.BYTE XDATA
016604	000220	XDATA=XDATA+1
	220	.BYTE XDATA
016605	000221	XDATA=XDATA+1
	221	.BYTE XDATA
016606	000222	XDATA=XDATA+1
	222	.BYTE XDATA
016607	000223	XDATA=XDATA+1
	223	.BYTE XDATA
016610	000224	XDATA=XDATA+1
	224	.BYTE XDATA
016611	000225	XDATA=XDATA+1
	225	.BYTE XDATA
016612	000226	XDATA=XDATA+1
	226	.BYTE XDATA
016613	000227	XDATA=XDATA+1
	227	.BYTE XDATA
016614	000230	XDATA=XDATA+1
	230	.BYTE XDATA
016615	000231	XDATA=XDATA+1
	231	.BYTE XDATA
016616	000232	XDATA=XDATA+1
	232	.BYTE XDATA
016617	000233	XDATA=XDATA+1
	233	.BYTE XDATA
016620	000234	XDATA=XDATA+1
	234	.BYTE XDATA
016621	000235	XDATA=XDATA+1
	235	.BYTE XDATA
016622	000236	XDATA=XDATA+1
	236	.BYTE XDATA
016623	000237	XDATA=XDATA+1
	237	.BYTE XDATA
016624	000240	XDATA=XDATA+1
	240	.BYTE XDATA
016625	000241	XDATA=XDATA+1
	241	.BYTE XDATA
016626	000242	XDATA=XDATA+1
	242	.BYTE XDATA
016627	000243	XDATA=XDATA+1
	243	.BYTE XDATA
016630	000244	XDATA=XDATA+1
	244	.BYTE XDATA
016631	000245	XDATA=XDATA+1
	245	.BYTE XDATA
016632	000246	XDATA=XDATA+1
	246	.BYTE XDATA
016633	000247	XDATA=XDATA+1
	247	.BYTE XDATA
	000250	XDATA=XDATA+1

016634	250	.BYTE XDATA
	000251	XDATA=XDATA+1
016635	251	.BYTE XDATA
	000252	XDATA=XDATA+1
016636	252	.BYTE XDATA
	000253	XDATA=XDATA+1
016637	253	.BYTE XDATA
	000254	XDATA=XDATA+1
016640	254	.BYTE XDATA
	000255	XDATA=XDATA+1
016641	255	.BYTE XDATA
	000256	XDATA=XDATA+1
016642	256	.BYTE XDATA
	000257	XDATA=XDATA+1
016643	257	.BYTE XDATA
	000260	XDATA=XDATA+1
016644	260	.BYTE XDATA
	000261	XDATA=XDATA+1
016645	261	.BYTE XDATA
	000262	XDATA=XDATA+1
016646	262	.BYTE XDATA
	000263	XDATA=XDATA+1
0 547	263	.BYTE XDATA
	000264	XDATA=XDATA+1
016650	264	.BYTE XDATA
	000265	XDATA=XDATA+1
016651	265	.BYTE XDATA
	000266	XDATA=XDATA+1
016652	266	.BYTE XDATA
	000267	XDATA=XDATA+1
016653	267	.BYTE XDATA
	000270	XDATA=XDATA+1
016654	270	.BYTE XDATA
	000271	XDATA=XDATA+1
016655	271	.BYTE XDATA
	000272	XDATA=XDATA+1
016656	272	.BYTE XDATA
	000273	XDATA=XDATA+1
016657	273	.BYTE XDATA
	000274	XDATA=XDATA+1
016660	274	.BYTE XDATA
	000275	XDATA=XDATA+1
016661	275	.BYTE XDATA
	000276	XDATA=XDATA+1
016662	276	.BYTE XDATA
	000277	XDATA=XDATA+1
016663	277	.BYTE XDATA
	000300	XDATA=XDATA+1
016664	300	.BYTE XDATA
	000301	XDATA=XDATA+1
016665	301	.BYTE XDATA
	000302	XDATA=XDATA+1
016666	302	.BYTE XDATA
	000303	XDATA=XDATA+1
016667	303	.BYTE XDATA
	000304	XDATA=XDATA+1
016670	304	.BYTE XDATA

016671	000305	XDATA=XDATA+1
	305	.BYTE XDATA
016672	000306	XDATA=XDATA+1
	306	.BYTE XDATA
016673	000307	XDATA=XDATA+1
	307	.BYTE XDATA
016674	000310	XDATA=XDATA+1
	310	.BYTE XDATA
016675	000311	XDATA=XDATA+1
	311	.BYTE XDATA
016676	000312	XDATA=XDATA+1
	312	.BYTE XDATA
016677	000313	XDATA=XDATA+1
	313	.BYTE XDATA
016700	000314	XDATA=XDATA+1
	314	.BYTE XDATA
016701	000315	XDATA=XDATA+1
	315	.BYTE XDATA
016702	000316	XDATA=XDATA+1
	316	.BYTE XDATA
016703	000317	XDATA=XDATA+1
	317	.BYTE XDATA
016704	000320	XDATA=XDATA+1
	320	.BYTE XDATA
016705	000321	XDATA=XDATA+1
	321	.BYTE XDATA
016706	000322	XDATA=XDATA+1
	322	.BYTE XDATA
016707	000323	XDATA=XDATA+1
	323	.BYTE XDATA
016710	000324	XDATA=XDATA+1
	324	.BYTE XDATA
016711	000325	XDATA=XDATA+1
	325	.BYTE XDATA
016712	000326	XDATA=XDATA+1
	326	.BYTE XDATA
016713	000327	XDATA=XDATA+1
	327	.BYTE XDATA
016714	000330	XDATA=XDATA+1
	330	.BYTE XDATA
016715	000331	XDATA=XDATA+1
	331	.BYTE XDATA
016716	000332	XDATA=XDATA+1
	332	.BYTE XDATA
016717	000333	XDATA=XDATA+1
	333	.BYTE XDATA
016720	000334	XDATA=XDATA+1
	334	.BYTE XDATA
016721	000335	XDATA=XDATA+1
	335	.BYTE XDATA
016722	000336	XDATA=XDATA+1
	336	.BYTE XDATA
016723	000337	XDATA=XDATA+1
	337	.BYTE XDATA
016724	000340	XDATA=XDATA+1
	340	.BYTE XDATA
	000341	XDATA=XDATA+1

016725	341	.BYTE XDATA
	000342	XDATA=XDATA+1
016726	342	.BYTE XDATA
	000343	XDATA=XDATA+1
016727	343	.BYTE XDATA
	000344	XDATA=XDATA+1
016730	344	.BYTE XDATA
	000345	XDATA=XDATA+1
016731	345	.BYTE XDATA
	000346	XDATA=XDATA+1
016732	346	.BYTE XDATA
	000347	XDATA=XDATA+1
016733	347	.BYTE XDATA
	000350	XDATA=XDATA+1
016734	350	.BYTE XDATA
	000351	XDATA=XDATA+1
016735	351	.BYTE XDATA
	000352	XDATA=XDATA+1
016736	352	.BYTE XDATA
	000353	XDATA=XDATA+1
016737	353	.BYTE XDATA
	000354	XDATA=XDATA+1
016740	354	.BYTE XDATA
	000355	XDATA=XDATA+1
016741	355	.BYTE XDATA
	000356	XDATA=XDATA+1
016742	356	.BYTE XDATA
	000357	XDATA=XDATA+1
016743	357	.BYTE XDATA
	000360	XDATA=XDATA+1
016744	360	.BYTE XDATA
	000361	XDATA=XDATA+1
016745	361	.BYTE XDATA
	000362	XDATA=XDATA+1
016746	362	.BYTE XDATA
	000363	XDATA=XDATA+1
016747	363	.BYTE XDATA
	000364	XDATA=XDATA+1
016750	364	.BYTE XDATA
	000365	XDATA=XDATA+1
016751	365	.BYTE XDATA
	000366	XDATA=XDATA+1
016752	366	.BYTE XDATA
	000367	XDATA=XDATA+1
016753	367	.BYTE XDATA
	000370	XDATA=XDATA+1
016754	370	.BYTE XDATA
	000371	XDATA=XDATA+1
016755	371	.BYTE XDATA
	000372	XDATA=XDATA+1
016756	372	.BYTE XDATA
	000373	XDATA=XDATA+1
016757	373	.BYTE XDATA
	000374	XDATA=XDATA+1
016760	374	.BYTE XDATA
	000375	XDATA=XDATA+1
016761	375	.BYTE XDATA

```

016762 000376          XDATA=XDATA+1
          376          .BYTE  XDATA
016763 000377          XDATA=XDATA+1
          377          .BYTE  XDATA
          000400       XDATA=XDATA+1
7          .EVEN
8 016764          .PFAIL
                                ;ENTER HERE ON POWER FAILURE

016764 010046          PFAIL:  MOV    R0,-(SP)          ;SAVE R0 R5 ON PROCESSOR STACK
016766 010146          MOV    R1,-(SP)
016770 010246          MOV    R2,-(SP)
016772 010346          MOV    R3,-(SP)
016774 010446          MOV    R4,-(SP)
016776 010546          MOV    R5,-(SP)
017000 016746 161020   MOV    24,-(SP)
017004 010667 177334   MOV    SP,SAVSP          ;SAVE STACK POINTER
017010 012767 017022 161006  MOV    @RESTART,24      ;SET UP FOR POWER UP TRAP
017016 000000          HALT                                ;HALT ON POWER DOWN NORMAL
017020 000777          BR     .                                ; 3

                                ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

017022 016706 177316   RESTAR: MOV    SAVSP,SP          ;RESTORE STACK POINTER
017026 012605          MOV    (SP)+,R5          ;RESTORE R0-R5
017030 012604          MOV    (SP)+,R4
017032 012603          MOV    (SP)+,R3
017034 012602          MOV    (SP)+,R2
017036 012601          MOV    (SP)+,R1
017040 012600          MOV    (SP)+,R0
017042 012767 016764 160754  MOV    @PFAIL,24          ;SET UP FOR POWER FAILURE
017050 012767 000340 160720  MOV    #340,PS
017056 012706 020164   MOV    @STACK,SP
017062 005067 000370   CLR    TEMP
017066 005267 000364   INC    TEMP
017072 001375          BNE    .-4
017074 104401          TYPE
017076 017257          MCRLF                                ; 5
017100 104402          OCTASC                                ; 5
017102 017124          PFTAB
017104 104401          TYPE
017106 017262          MPFAIL
017110 005067 177174   CLR    ERRFLG
017114 005067 177234   CLR    LAST
017120 000177 177172   JMP    @RETRN
017124 000001          PFTAB:  1
017126 000006 000002   6,2
017132 016316          RETRN

9 017134          .MSG  +/DH11 SINGLE LINE DATA TEST/,+/CZDMF-CO/
017134          015      012      012  MTITLE:  .ASCIZ  <15><12><12>/DH11 SINGLE LINE DATA TEST /<15><12>
017137          104      110      061
017142          061      040      123
017145          111      116      107
017150          114      105      040
017153          114      111      116
017156          105      040      104

```

017161	101	124	101	
017164	040	124	105	
017167	123	124	040	
017172	015	012	000	
017175	015	012	126	MVECTOR: .ASCIZ <15><12>/VECTOR ADDRESS-/
017200	105	103	124	
017203	117	122	040	
017206	101	104	104	
017211	122	105	123	
017214	123	055	000	
017217	015	012	103	MREGAD: .ASCIZ <15><12>/CONTROL REGISTER ADDRESS-/
017222	117	116	124	
017225	122	117	114	
017230	040	122	105	
017233	107	111	123	
017236	124	105	122	
017241	040	101	104	
017244	104	122	105	
017247	123	123	055	
017252	000			
017253	040	040	077	MQM: .ASCIZ / ?/
017256	000			
017257	015	012	000	MCRLF: .ASCIZ <15><12>
017262	040	040	120	MPFAIL: .ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
017265	117	127	105	
017270	122	040	106	
017273	101	111	114	
017276	125	122	105	
017301	054	040	120	
017304	122	117	107	
017307	122	101	115	
017312	040	122	105	
017315	123	124	101	
017320	122	124	040	
017323	101	124	040	
017326	124	105	123	
017331	124	040	111	
017334	116	040	120	
017337	122	117	107	
017342	122	105	123	
017345	123	000		
017347	015	012	103	MEPASS: .ASCIZ <15><12>/CZDHF-CO/
017352	132	104	110	
017355	106	055	103	
017360	060	000		
017362	015	012	120	PASTXT: .ASCIZ <15><12>/PASS COUNT = /
017365	101	123	123	
017370	040	103	117	
017373	125	116	124	
017376	040	075	040	
017401	000			
017402	015	012	122	MR: .ASCIZ <15><12>/R/
017405	000			
017406	015	012	124	MTSTPC: .ASCIZ <15><12>/TEST PC-/
017411	105	123	124	
017414	040	120	103	

: 5


```

017417      055      000
10
11 017422      .EVEN
      .EVEN
      .TRPTAB

      ;TABLE OF POINTERS FOR TRAP DECODING

017422      015100      TRPTAB: SCOPER
017424      015554      TYPER
017426      016074      OCTASN
017430      015606      INSTRG
017432      015700      INSTRE
017434      015710      PARAMS
017436      015462      SVOSP
017440      015522      RSOS
017442      015200      SCOP1R
12 017444      .BUFFER

      ;BUFFERS FOR INPUT-OUTPUT

017444      000000      INBUF:  0
      017456      . = .+10
017456      000000      TEMP:   0
      017470      . = .+10

017470      000000      MDATA:  0
      017502      . = .+10
13 017502      .ERRTAB

      ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA

017502      ERRTAB:
14 017502      017516      EM1
15 017504      017702      DT1
16 017506      017557      CM2
17 017510      017720      DT2
18 017512      017630      EM3
19 017514      017742      DT3
20 017516      104      101      124      EM1:  .ASCIZ  /DATA ERROR/<15><12>/EXP      REC      LINE/
      017521      101      040      105
      017524      122      122      117
      017527      122      015      012
      017532      105      130      120
      017535      040      040      040
      017540      040      040      122
      017543      105      103      040
      017546      040      040      040
      017551      040      114      111
      017554      116      105      000
21 017557      104      101      124      EM2:  .ASCIZ  /DATA ERROR/<15><12>/EXP      REC      SPEED      LINE/
      017562      101      040      105
      017565      122      122      117
      017570      122      015      012
      017573      105      130      120
      017576      040      040      040
      017601      040      040      122
      017604      105      103      040

```

Line	Address	Hex	Hex	Hex	Text	REC	LENGTH	LINE/
	017607	040	040	040				
	017612	040	123	120				
	017615	105	105	104				
	017620	040	040	040				
	017623	114	111	116				
	017626	105	000					
22	017630	104	101	124	EM3: .ASCIZ /DATA ERROR/<15><12>/EXP			
	017633	101	040	105				
	017636	122	122	117				
	017641	122	015	012				
	017644	105	130	120				
	017647	040	040	040				
	017652	040	040	122				
	017655	105	103	040				
	017660	040	040	040				
	017663	040	114	105				
	017666	116	107	124				
	017671	110	040	040				
	017674	114	111	116				
	017677	105	000					
23					.EVEN			
24	017702	000003			DT1: 3			
25	017704	006	002		.BYTE 6.2			
26	017706	016356			TDATA			
27	017710	006	002		.BYTE 6.2			
28	017712	016340			SAVR4			
29	017714	002	000		.BYTE 2.0			
30	017716	016336			SAVR3			
31	017720	000004			DT2: 4			
32	017722	006	002		.BYTE 6.2			
33	017724	016360			RDATA			
34	017726	006	002		.BYTE 6.2			
35	017730	016336			SAVR3			
36	017732	002	005		.BYTE 2.5			
37	017734	016334			SAVR2			
38	017736	002	000		.BYTE 2.0			
39	017740	016342			SAVR5			
40	017742	000004			DT3: 4			
41	017744	006	002		.BYTE 6.2			
42	017746	016360			RDATA			
43	017750	006	002		.BYTE 6.2			
44	017752	016336			SAVR3			
45	017754	002	006		.BYTE 2.6			
46	017756	016340			SAVR4			
47	017760	002	000		.BYTE 2.0			
48	017762	016342			SAVR5			
49	017764				.ENDCOD			
	017764	000000			ENDCOD: 0			
50		000001			.END			

ADRCNT = 016073	EOP 015000	PARAM = 104405	SW00 = 000001	T24 005232
BEGIN 001306	ERRCNT 016314	PARAMS 015710	SW01 = 000002	T25 005430
BINWRD 016244	ERRFLG 016310	PARAM1 015740	SW02 = 000004	T26 005626
BITX = 000000	ERRMSG 015336	PARERR 016014	SW03 = 000010	T27 006024
BIT00 = 000001	ERRORS 015216	PASARG 015072	SW04 = 000020	T3 001704
BIT01 = 000002	ERRTAB 017502	PASCNT 016312	SW05 = 000040	T30 006222
BIT02 = 000004	ERTAB0 015422	PASTXT 017362	SW06 = 000100	T31 006420
BIT03 = 000010	ESCAPE 016320	PFAIL 016764	SW08 = 000400	T32 006616
BIT04 = 000020	EXITER 015400	PFTAB 017124	SW09 = 001000	T33 007014
BIT05 = 000040	FREEZ1 016322	POPRO = 012600	SW10 = 002000	T34 007212
BIT06 = 000100	HALTS 015360	POP1SP = 005726	SW11 = 004000	T35 007410
BIT07 = 000200	HILIM 016066	POP2SP = 022626	SW12 = 010000	T36 007606
BIT08 = 000400	ICOUNT 016324	PS = 1776	SW13 = 020000	T37 010004
BIT09 = 001000	INBUF 017444	PUSHRO = 010046	SW14 = 040000	T4 002046
BIT10 = 002000	INIFLG 016350	PUSH1S = 005746	SW15 = 100000	T40 010202
BIT11 = 004000	INSTER = 104404	PUSH2S = 024646	TBUF 016364	T41 010400
BIT12 = 010000	INSTR = 104403	RDATA 016360	TDATA 016356	T42 010620
BIT13 = 020000	INSTRE 015700	RESREG 015356	TEMP 017456	T43 011040
BIT14 = 040000	INSTRG 015606	RESTAR 017022	TKCSR 016246	T44 011260
BIT15 = 100000	INSTR1 015620	RESTRT 015066	TKDBR 016250	T45 011500
BYICNT 016362	INSTR2 015706	RES05 = 104407	TPCSR 016252	T46 011720
CHRCNT 016242	LAST 016354	RETRN 016316	TPDBR 016254	T47 012140
DATABP 015354	LIGHTS 001002	RS05 015522	TRPOK 015442	T5 002210
DEVADR 016070	LIMITS 016020	SAVPC 016346	TRPSRV 015430	T50 012360
DHBA 016264	LINE = 000020	SAVR0 016330	TRPTAB 017422	T51 012600
DHBAR 016270	LOBITS 016072	SAVR1 016332	TYPDAT 015540	T52 013020
DHBC 016266	LOGICA 015056	SAVR2 016334	TYPE = 104401	T53 013240
DHBCR 016272	LOLIM 016064	SAVR3 016336	TYPER 015554	T54 013460
DHLPR 016262	LPCNT 016326	SAVR4 016340	TYPMSG 015316	T55 013700
DHNRC 016260	MCRLF 017257	SAVR5 016342	T1 001400	T56 014120
DHRLVL 016302	MDATA 017470	SAVSP 016344	T10 002656	T57 014340
DHRVEC 016300	MEPASS 017347	SAV0SP = 104406	T11 003020	T6 002352
DHSCR 016256	MPFAIL 017262	SCOPE = 104400	T12 003162	T60 014560
DHSLR 016276	MQM 017253	SCOPE1 = 015100	T13 003324	T7 002514
DHSR 016274	MR 017402	SCOPE1 = 104410	T14 003466	VEC1 001164
DHTLVL 016306	MREGAD 017217	SCOPE1R = 015200	T15 003630	VEC2 001174
DHTVEC 016304	MSG 015622	SPACNT = 016243	T16 003772	WRDCNT 016240
DT1 017702	MTITLE 017134	STACK = 020164	T17 004134	X = 000000
DT2 017720	MTSTPC 017406	START 001004	T2 001542	XBIT = 000001
DT3 017742	MVECTO 017175	STFLG 016352	T20 004276	XDATA = 000400
EM1 017516	N = 000001	SV05 015470	T21 004440	XLINE = 000000
EM2 017557	OCTASC = 104402	SV05P 015462	T22 004636	XN = 000061
EM3 017630	OCTASN 016074	SWR 001000	T23 005034	Y = 000011
ENDCOD 017764				

. ABS. 017766 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 19200 WORDS (75 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 71 PAGES
CZDHF.C.BIN,CZDHF.C.SEQ=CZDHF.C.DOC,DHMACA.MAC,CZDHF.C.P11