DH11

BREAK AND HALF DOP

MD-11-DZDHI-C

EP-DZDHI-C-DL-B

DEC 1976

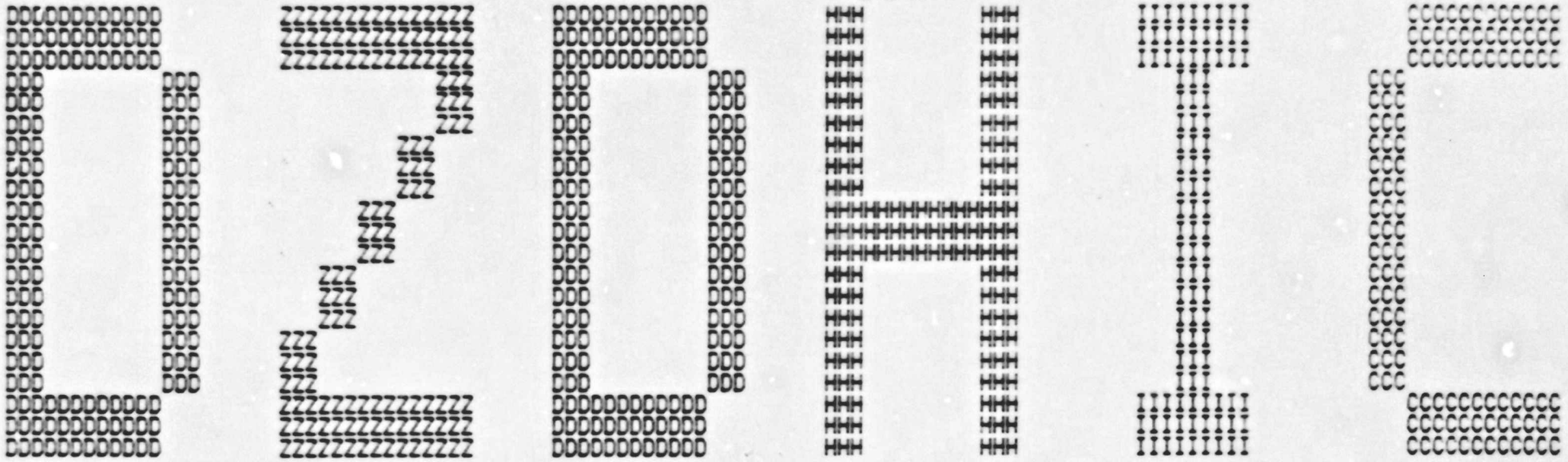
digital

FICHE 1 OF 1

MADE IN USA

```
3PEE7896123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012LPT
[0ESTART+ADar DAVIES,TOM[400,2264] 3555 DZDHIC STAITI1 D000 03MDecM76 09:50:37 Monitor IPC-D 602 [680] *START*
D   D  A   A  V   V    I     E     S           T    O   O  MM MM
D   D  A   A  V   V    I     E     S           T    O   O  M M M
D   D  A   A  V   V    I     EEEE   SSS        T    O   O  M   M
D   D  AAAAA  V   V    I     E         S       T    O   O  M   M
D   D  A   A   V V     I     E         S       T    O   O  M   M
DDDD   A   A    V     III    EEEEE  SSSS   ,   T     000   M   M


MTO:ML21-4:DAVIES2-- distribution to MH21-4, slot 134     1    333   4   4     >
MM MM  L      2   2    11               4   4        <   11   3   3  4   4      >
M M M  L          2     1               4   4       <     1       3  4   4       >
M   M  L        2       1     -----     44444      <      1      3   44444        >
M   M  L      2         1                   4       <     1       3      4       >
M   M  L    2           1                   4        <    1   3   3      4      >
M   M  LLLLL  22222    111                  4         <  111   333       4     >


/[redacted]-UeDAV[redacted]IES,TOM[400,2264]eJML21[redacted]DHIC0Se@3411 Date 03-Dec-76 09:50:37 Monitor IPC-D 602 [680] *START*
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111111111111110
0000000011111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111111222222222233312
2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012 *S
D   D  A   A  V   V    I     E     S           T    O   O  MM MM
D   D  A   A  V   V    I     E     S           T    O   O  M M M
D   D  A   A  V   V    I     EEEE   SSS        T    O   O  M   M
D   D  AAAAA  V   V    I     E         S       T    O   O  M   M
D   D  A   A   V V     I     E         S   ''  T    O   O  M   M
DDDD   A   A    V     III    EEEEE  SSSS   ,   T     000   M   M


MTO:ML21-4:DAVIES2-- distribution to MH21-4, slot 134     1    333   4   4     >
MM MM  L      2   2    11               4   4        <   11   3   3  4   4      >
M M M  L          2     1               4   4       <     1       3  4   4       >
M   M  L        2       1     -----     44444      <      1      3   44444        >
M   M  L      2         1                   4       <     1       3      4       >
M   M  L    2           1                   4        <    1   3   3      4      >
M   M  LLLLL  22222    111                  4         <  111   333       4     >


/[redacted]-UeDAV[redacted]IES,TOM[400,2264]eJML21[redacted]DHIC0Se@3411 Date 03-Dec-76 09:50:37 Monitor IPC-D 602 [680] *START*
```

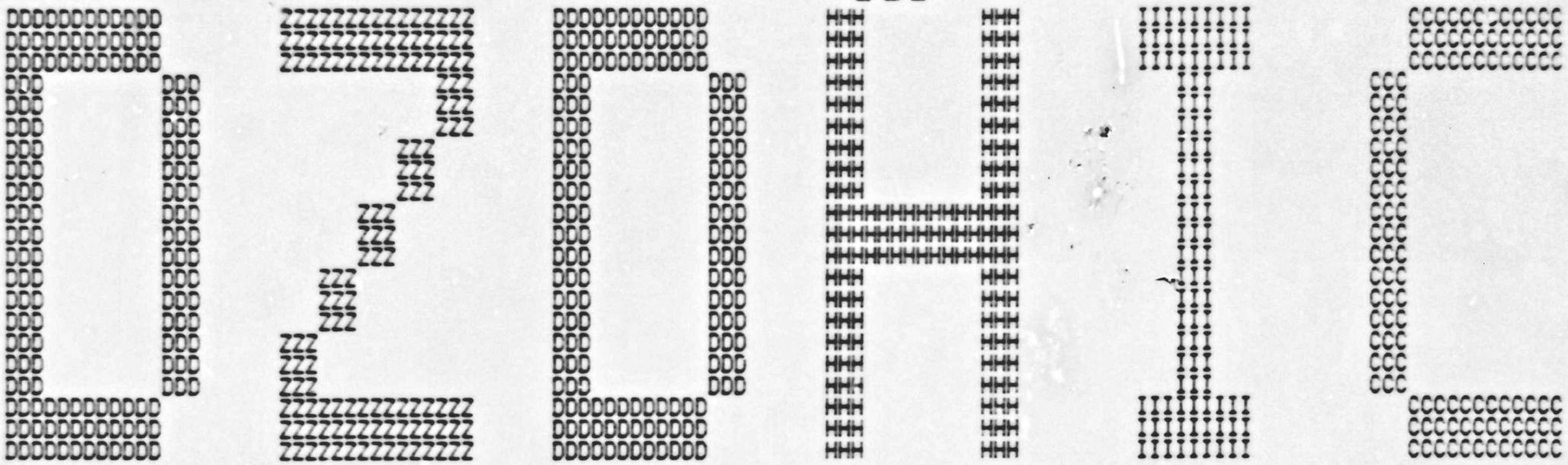
C01

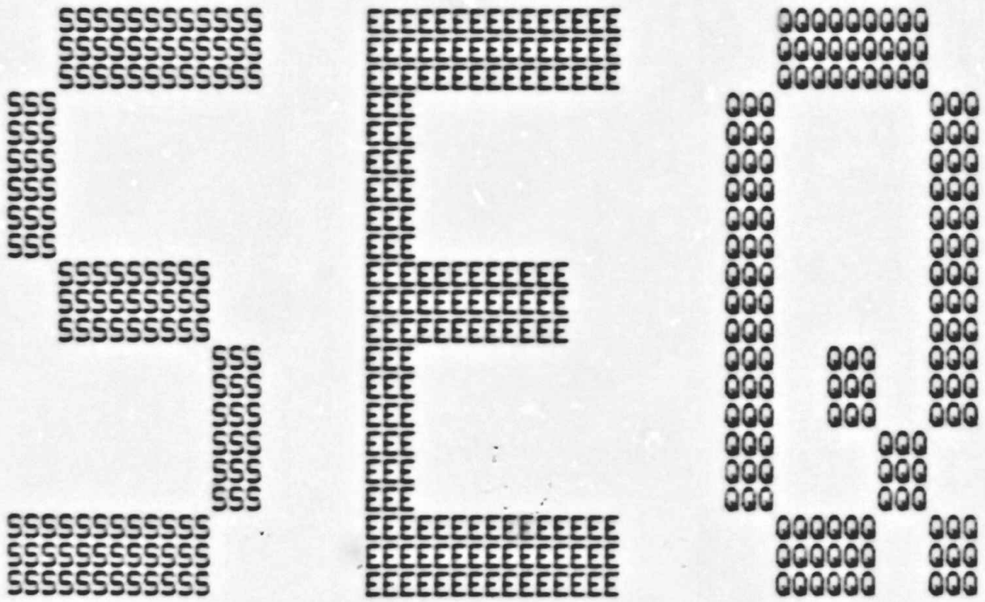


LPTSPL Version 6(100344) Running on MTA041
 START User DAVIES,TOM [400,2704] Job DZDHIC Seq. 11 Date 03-Dec-76 09:50:37 Monitor IPC-D 602 [680] *START*
Request created: 03-Dec-76 09:26:29   /TO:ML21-4:DAVIES -- distribution to ML21-4, slot 134

[illegible] 09:50:41
File will be RENAMEd to <057> protection

D01





LPTSPL Version 6(100344) Running on MTA041
START User DAVIES,TOM [400,2704] Job DZDHIC Seq. 11 Date 03-Dec-76 09:50:37 Monitor IPC-D 602 [680] *START*
Request created: 03-Dec-76 09:26:29 /TO:ML21-4:DAVIES -- distribution to ML21-4, slot 134

[illegible] 09:50:44
File will be RENAMEd to <057> protection

.REM !

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDHI-C-D

PRODUCT NAME: DH11 BREAK AND HALF-DUPLEX TEST

DATE: DECEMBER 1976

MAINTAINER: DIAGNOSTIC GROUP

1. ABSTRACT

THE DH11 BREAK AND HALF DUPLEX TEST CHECKS THE BREAK
CONTROL LOGIC OF THE DH11 AND VERIFIES THAT THE UARTS
RECEIVE ONLY ONE BREAK CHARACTER ON A GIVEN
LINE NO MATTER HOW LONG BREAK IS ASSERTED.
THE TEST ALSO VERIFIES THAT NO CHARACTERS ARE
RECEIVED ON A A LINE IF THE HALF DUPLEX FUNCTION FOR
THAT LINE IS SELECTED.

```
                        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 BREAK AND HALF-DUPLEX TEST"
                                AND WILL THEN TYPE "VECTOR ADDRESS-" AND WAIT FOR AN
                                INPUT FROM THE TELETYPE KEYBOARD.
```

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 BREAK AND HALF-DUPLEX 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 BREAK AND HALF-DUPLEX 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 THS FEATURE IS USED, SINCE
THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS THAT
IS IN THE MIDDLE OF A TEST

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

```
                        5.      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.
```

```
223
224                             5.2.4   EOP     (END OF PASS)
225
226                                     THIS ROUTINE IS ENTERED ONCE PER PASS AFTER ALL TESTS
227                                     HAVE BEEN COMPLETED.  THIS ROUTINE TYPES THE MAINDEC
228                                     IDENTIFICATION CODE OF THE PROGRAM, CLEARS ERROR FLAGS
229                                     AND UPDATES THE PASS COUNT.  IF THE PROGRAM WAS LOADED
230                                     UNDER ACT11 OR DDP, THE ROUTINE CHECKS FOR RETURN TO
231                                     THE ACT11 OR DDP MONITOR.  IF THE PROGRAM IS NOT UNDER
232                                     MONITOR CONTROL, THE ROUTINE TRANSFERS TO BEGIN.
233
234                             5.2.5   SCOPER  (SCOPE LOOP AND ITERATION HANDLER)
235
236                                     THIS ROUTINE IS ENTERED EACH TIME A TEST IS COMPLETED.
237                                     THE ROUTINE CHECKS FOR THE FOLLOWING UPON ENTRY
238                                     A) IF SW10=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST
239                                        IN SEQUENCE, AFTER CLEARING ERROR FLAGS.
240                                     B) IF SW11=1, THE ROUTINE WILL TRANSFER TO THE NEXT TEST
241                                        SEQUENCE, AFTER CLEARING ERROR FLAGS.
242                                     C) IF SW14=1, THE ROUTINE WILL LOOP ON THE CURRENT TEST
243                                        REGARDLESS OF THE ITERATION COUNT.
244
245                                     IF NONE OF THE ABOVE IS TRUE, THE ROUTINE WILL ADD 1
246                                     TO THE COUNT OF TEST ITERATIONS, AND COMPARE THIS VALUE
247                                     TO THE NUMBER OF ITERATIONS THAT SHOULD BE PERFORMED.
248                                     IF THESE NUMBERS ARE EQUAL, THE ROUTINE WILL TRANSFER
249                                     TO THE NEXT TEST IN SEQUENCE.  THE THE NUMBERS ARE NOT
250                                     EQUAL, THE TEST CURRENTLY IN PROGRESS WILL BE REPEATED.
251
252                             5.2.6   SCOP1R  (FREEZE ON CURRENT DATA)
253
254                                     THE CALL TO THIS ROUTINE FOLLOWS IMMEDIATLY AFTER THE
255                                     THE CALL TO THE ERROR HANDLER IN THOSE TESTS THAT HAVE
256                                     VARIABLE PARAMETERS.  THIS ROUTINE IS ALWAYS ENTERED IN
257                                     THOSE TESTS, WHETHER OR NOT AN ERROR OCCURS.
258                                     IF SW09=1, THE ROUTINE WILL TRANSFER CONTROL BACK TO THE
259                                     TEST AT A POINT WHICH WILL ALLOW REPEATING THE FUNCTION
260                                     UNDER TEST CONTINUOUSLY WITH THE SAME DATA.  IF THIS
261                                     OPTION IS SELECTED, THE ROUTINE "SCOPER" IS NEVER ENTERED
262                                     AND ITERATION COUNTS WILL NOT BE UPDATED.
```

```
263
264                             5.2.7   ERRORS  (ERROR HANDLER)
265
266                                     THIS ROUTINE IS ENTERED UPON ERROR DETECTION ONLY.
267                                     WITH ALL CONSOLE SWITCHES DOWN, THE ROUTINE PROCEDES AS FOLLOWS:
268                                     A) THE PC OF THE INSTRUCTION THAT CALLED THE ERROR HANDLER
269                                        IS ACCESSED THRU THE STACK, AND THEN THE EMT INSTRUCTION
270                                        ITSELF IS FETCHED.  THE 8 LSB OF THE EMT
271                                        INSTRUCTION ARE THE ERROR CODE.  THIS CODE IS
272                                        USED TO ACCESS A TABLE OF ERROR MESSAGES AND ERROR
273                                        DATA STORAGE LOCATIONS.
274                                     B) IF THE TEST THAT FAILED DID NOT FAIL PREVIOUSLY
275                                        DURING THIS PASS,  A COMPLETE ERROR REPORT IS MADE
276                                        IF THE TEST THAT FAILED FAILED MOR THAT ONCE DURING
277                                        THE CURRENT PASS, ONLY THE DATA RELATING TO THE FAILUER
278                                        IS TYPED.  IF SW13=1, NO ERROR TYPEOUT IS MADE.
279                                     C) THE ROUTINE NOW CHECKS FOR HALT ON ERROR.  IF SW15=1
280                                        THE PROGRAM WILL HALT WITH THE PC OF THE CALL TO
281                                        THE ERROR ROUTINE IN R0.  IF SW15=0, THE PROGRAM WILL
282                                        NOT HALT, BUT WILL CHECK FOR ESCAPE TO NEXT TEST.
283                                     D) IF SW10=0, THE ROUTINE WILL RETURN
284                                        TO THE TEST IN PROGRESS.  IF SW10=1, THE ROUTINE WILL
285                                        ABORT THE CURRENT TEST, AND TRANSFER TO THE NEXT
286                                        TEST IN SEQUENCE, THRU THE ROUTINE "SCOPER".
287
288                             5.2.8   TRPSRV  (TRAP DECODE AND DISPATCH)
289
290                                     THIS ROUTINE DECODES THE 8 LSB OF THE TRAP INSTRUCTION
291                                     THAT CAUSED TH PROGRAM INTERRUPT, AND TRANSFERS CONTROL
292                                     TO THE ROUTINE THRU THE TABLE "TRPTAB" USING THE 8 LSB
293                                     OF THE TRAP INSTRUCTION AS  AN OFFSET TO THE POINTER TO
294                                     THE ROUTINE TO BE ENTERED.
295
```

```
296
297                         5.3     PROGRAM AND OR OPERATOR ACTION
298
299                         5.3.1   PROGRAM START WITH ALL SWITCHES DOWN
300
301                         5.3.1.1 REFER TO SECTIONS 4.3.1 AND 4.3.2 FOR INITIAL PROGRAM
302                                 BEHAVIOR.
303
304                         5.3.1.2 AFTER "R" HAS BEEN TYPED BY THE PROGRAM, TEST EXECUTION
305                                 WILL BEGIN.  EACH TEST WILL BE REPEATED A SELECTED NUMBER
306                                 OF ITERATIONS (SEE LISTING FOR EXACT NUMBER FOR EACH TEST)
307                                 AND THEN THE PROGRAM WILL PROCEED TO THE NEXT TEST.
308
309                         5.3.1.3 WHEN ALL ITERATIONS HAVE BEEN COMPLETED, THE PROGRAM
310                                 WILL TYPE "DZDHI" AND THEN RESTART TESTING AT TEST 1
311                                 (LOCATION T1 IN THE PROGRAM).
312
313                         5.3.1.4 IF AN ERROR OCCURS, THE PROGRAM WILL TYPE AN APPROPRIATE
314                                 ERROR MESSAGE, AND THEN CONTINUE THE TEST IN PROGRESS.
315
316                         5.3.2   PROGRAM START WITH SW00=1
317
318                                 THE PROGRAM WILL PERFORM AS DESCRIBED IN 4.3.1 AND 5.3.1
319
320                         5.3.3   PROGRAM START WITH SW01=1
321
322                         5.3.3.1 REFER TO SECTION 4.3.4 FOR INITIAL PROGRAM BEHAVIOR
323
324                         5.3.3.2 TEST EXECUTION WILL START AT THE ADDRESS SPECIFIED
325                                 AND WILL CONTINUE AS DESCRIBED IN 5.3.1.2
326
327                         5.3.3.3 AFTER "DZDHI" HAS BEEN TYPED, THE PROGRAM WILL
328                                 RESUME TESTING AT TEST 1
329
330                         5.3.4   PROGRAM OPERATION WITH SW15=1
331
332                                 SAME AS 5.3.1, EXCEPT THAT IN THE CASE OF AN ERROR,
333                                 THE PROGRAM WILL HALT AFTER THE ERROR TYPEOUT, AND
334                                 THE PC+2 OF THE CALL TO THE ERROR ROUTINE WILL BE
335                                 DISPLAYED IN R0.
336
337                         5.3.5   PROGRAM OPERATION WITH SW13=1
338
339                                 SAME AS 5.3.1 EXCEPT THAT NO ERROR TYPEOUTS WILL OCCUR
340
341                         5.3.6   PROGRAM OPERATION WITH SW11=1
342
343                                 SAME AS 5.3.1 EXCEPT THAT EACH TEST WILL BE REPEATED ONCE
344                                 ONLY
345
346                         5.3.7   PROGRAM OPERATION WITH SW10=1
347
348                                 SAME AS 5.3.1, EXCEPT THAT IN THE CASE OF AN ERROR
349                                 THE CURRENT TEST WILL BE ABORTED, AND THE PROGRAM
350                                 WILL PROCEED TO THE NEXT TEST IN SEQUENCE.
```

```
                        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 OPERATOT ACTION IS
                                REQUIRED TO CONTINUE TESTING
                        6.2.2   SW15=1

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

                        6.2.3   ILLEGAL INTERRUPTS

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

```
                        6.3     SCOPE LOOPING

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

                        6.3.2   TO SCOPE ON A SPECIFIC VALUE OF A 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 DZDHI TO END OF TYPEOUT OF DZDHI)
                                IS GIVEN FOR VARIOUS PROCESSORS IN THE TABLE BELOW

                                                TIME
                                PROCESSOR

                                PDP-11/05,10

                                PDP-11/20

                                PDP-11/40

                                PDP-11/45
```

```
                                9.      PROGRAM DESCRIPTION

                                        THE FIRST GROUP OF TESTS VERIFIES THAT ONLY ONE BREAK
                                        CHARACTER IS RECEIVED ON A GIVEN LINE, EVEN THOUGH BREAK
                                        IS ASSRETED FOR THAT LINE FOR 400 (OCTAL) CHARACTER TIMES.
                                        THERE IS AN INDIVIDUAL TEST LOOP FOR EACH LINE.
                                        THE TEST BEGINS BY FLUSING EACH UART BY TRANSMITTING
                                        TO NULL (0) CHARACTERS.  THE BREAK BIT IS THEN SET
                                        FOR THE LINE TO BE TESTED, TRANSMISSION OF THE BINARY COUNT
                                        PATTERN IS STARTED.  THE SILO IS THEN CHECKED TO VERIFY
                                        THAT ONLY ONE CHARACTER WAS RECEIVED
                                        AND THAT IT WAS A BREAK CHARACTER.

                                        THE SECOND GROUP OF TESTS VERIFIES THAT NO CHARACTERS
                                        ARE RECEIVED ON A SELECTED LINE IF THE HALF DUPLEX BIT
                                        IS SET FOR THE LINE TO BE TESTED.  THERE IS AN INDIVIDUAL
                                        TEST LOOP FOR EACH LINE TO BE TESTED.  A BINARY COUNT
                                        PATTERN IS THEN TRANSMITTED ON THE LINE TO BE TESTED.
                                        WHEN ALL CHARACTERS HAVE BEEN TRANSMITTED, THE CHARACTER
                                        AVAILABLE FLAG IS TESTED TO DETERMINE IF ANY CHARACTERS HAVE
                                        BEEN RECEIVED.

                                10.     LISTING


                                        !

                                ;DH11 BREAK AND HALF DUPLEX TEST
                                ;COPYRIGHT 1972, 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE DH11 BREAK AND HALF DUPLEX 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 " DZDHI "
                                        ;AND THEN RESUM TESTING

                                ;SWITCH REGISTER OPTIONS

100000                          SW15=100000             ;=1,HALT ON ERROR
040000                          SW14=40000              ;=1,LOOP ON CURRENT TEST
020000                          SW13=20000              ;=1,INHIBIT ERROR TYPEOUT
010000                          SW12=10000
004000                          SW11=4000               ;=1,INHIBIT ITERATIONS
002000                          SW10=2000               ;=1,ESCAPE TO NEXT TEST ON ERROR
001000                          SW09=1000               ;=1,LOOP WITH CURRENT DATA
```

```
    508          000400                  SW08=400
    509          000100                  SW06=100
    510          000040                  SW05=40
    511          000020                  SW04=20
    512          000010                  SW03=10
    513          000004                  SW02=4
    514          000002                  SW01=2
    515          000001                  SW00=1
    516
```

```
;RESTART PROGRAM AT SELECTED TEST
;RESELECT VECTOR AND CONTROL REGISTER
;ADDRESS AFTER PROGRAM RESTART
```

```
                        ;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

177570                  SWR=177570      ;CONSOLE SWITCH REGISTER
177570                  LIGHTS=177570   ;PDP-11/45 DISPLAY REGISTER
177776                  PS=177776       ;PROCESSOR STATUS WORD
013540                  STACK=ENDCOD+200;START OF PROCESSOR STACK

                        ;INSTRUCTION DEFINITIONS

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                  PUSHR0=10046    ;SAVE R0 ON STACK
012600                  POPR0=12600     ;RESTORE R0 FROM STACK
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                  POP2SP=22626    ;INCREMENT STACK TWICE
                        .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL


100000                  BIT15=100000
040000                  BIT14=40000
020000                  BIT13=20000
010000                  BIT12=10000
004000                  BIT11=4000
002000                  BIT10=2000
001000                  BIT09=1000
000400                  BIT08=400
000200                  BIT07=200
000100                  BIT06=100
000040                  BIT05=40
000020                  BIT04=20
000010                  BIT03=10
000004                  BIT02=4
000002                  BIT01=2
000001                  BIT00=1
```

```
                                        ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
                        000000  .=0
000000  000002                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000002  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000004  000006                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000006  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000010  000012                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000012  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000014  000016                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000016  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000020  000022                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000022  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000024  000026                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000026  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000030  000032                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000032  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000034  000036                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000036  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000040  000042                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000042  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000044  000046                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000046  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000050  000052                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000052  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000054  000056                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000056  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000060  000062                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000062  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000064  000066                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000066  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000070  000072                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000072  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000074  000076                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000076  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000100  000102                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000102  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000104  000106                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000106  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000110  000112                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000112  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000114  000116                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000116  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000120  000122                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000122  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000124  000126                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000126  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000130  000132                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000132  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000134  000136                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000136  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000140  000142                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000142  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000144  000146                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000146  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
000150  000152                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000152  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
```

```
   620  000154  000156                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   621  000156  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   622  000160  000162                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   623  000162  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   624  000164  000166                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   625  000166  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   626  000170  000172                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   627  000172  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   628  000174  000176                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   629  000176  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   630  000200  000202                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   631  000202  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   632  000204  000206                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   633  000206  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   634  000210  000212                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   635  000212  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   636  000214  000216                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   637  000216  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   638  000220  000222                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   639  000222  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   640  000224  000226                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   641  000226  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   642  000230  000232                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   643  000232  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   644  000234  000236                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   645  000236  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   646  000240  000242                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   647  000242  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   648  000244  000246                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   649  000246  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   650  000250  000252                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   651  000252  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   652  000254  000256                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   653  000256  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   654  000260  000262                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   655  000262  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   656  000264  000266                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   657  000266  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   658  000270  000272                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   659  000272  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   660  000274  000276                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   661  000276  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   662  000300  000302                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   663  000302  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   664  000304  000306                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   665  000306  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   666  000310  000312                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   667  000312  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   668  000314  000316                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   669  000316  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   670  000320  000322                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   671  000322  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   672  000324  000326                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   673  000326  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
   674  000330  000332                        .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   675  000332  000000                        HALT         ;EXAMINE STACK TO FIND CAUSE
```

```
   676  000334  000336                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   677  000336  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   678  000340  000342                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   679  000342  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   680  000344  000346                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   681  000346  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   682  000350  000352                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   683  000352  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   684  000354  000356                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   685  000356  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   686  000360  000362                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   687  000362  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   688  000364  000366                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   689  000366  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   690  000370  000372                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   691  000372  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   692  000374  000376                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   693  000376  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   694  000400  000402                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   695  000402  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   696  000404  000406                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   697  000406  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   698  000410  000412                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   699  000412  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   700  000414  000416                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   701  000416  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   702  000420  000422                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   703  000422  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   704  000424  000426                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   705  000426  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   706  000430  000432                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   707  000432  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   708  000434  000436                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   709  000436  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   710  000440  000442                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   711  000442  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   712  000444  000446                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   713  000446  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   714  000450  000452                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   715  000452  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   716  000454  000456                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   717  000456  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   718  000460  000462                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   719  000462  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   720  000464  000466                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   721  000466  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   722  000470  000472                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   723  000472  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   724  000474  000476                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   725  000476  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   726  000500  000502                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   727  000502  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   728  000504  000506                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   729  000506  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   730  000510  000512                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   731  000512  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
```

```
    732  000514  000516                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    733  000516  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    734  000520  000522                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    735  000522  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    736  000524  000526                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    737  000526  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    738  000530  000532                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    739  000532  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    740  000534  000536                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    741  000536  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    742  000540  000542                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    743  000542  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    744  000544  000546                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    745  000546  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    746  000550  000552                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    747  000552  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    748  000554  000556                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    749  000556  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    750  000560  000562                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    751  000562  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    752  000564  000566                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    753  000566  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    754  000570  000572                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    755  000572  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    756  000574  000576                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    757  000576  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    758  000600  000602                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    759  000602  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    760  000604  000606                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    761  000606  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    762  000610  000612                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    763  000612  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    764  000614  000616                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    765  000616  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    766  000620  000622                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    767  000622  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    768  000624  000626                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    769  000626  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    770  000630  000632                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    771  000632  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    772  000634  000636                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    773  000636  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    774  000640  000642                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    775  000642  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    776  000644  000646                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    777  000646  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    778  000650  000652                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    779  000652  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    780  000654  000656                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    781  000656  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    782  000660  000662                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    783  000662  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    784  000664  000666                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    785  000666  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
    786  000670  000672                          .+2          ;UNEXPECTED TRAP TO THIS LOCATION
    787  000672  000000                          HALT         ;EXAMINE STACK TO FIND CAUSE
```

```
   788  000674  000676                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   789  000676  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   790  000700  000702                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   791  000702  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   792  000704  000706                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   793  000706  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   794  000710  000712                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   795  000712  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   796  000714  000716                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   797  000716  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   798  000720  000722                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   799  000722  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   800  000724  000726                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   801  000726  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   802  000730  000732                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   803  000732  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   804  000734  000736                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   805  000736  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   806  000740  000742                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   807  000742  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   808  000744  000746                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   809  000746  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   810  000750  000752                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   811  000752  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   812  000754  000756                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   813  000756  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   814  000760  000762                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   815  000762  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   816  000764  000766                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   817  000766  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   818  000770  000772                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   819  000772  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
   820  000774  000776                    .+2          ;UNEXPECTED TRAP TO THIS LOCATION
   821  000776  000000                    HALT         ;EXAMINE STACK TO FIND CAUSE
```

```
                                        ;STANDARD INTERRUPT VECTORS
                                        
                                        
                000024          .=24
000024  012456                  PFAIL                   ;POWER FAIL HANDLER
000026  000340                  340                     ;SERVICE AT LEVEL 7
000030  010564                  ERRORS                  ;ERROR HANDLER
000032  000340                  340                     ;SERVICE AT LEVEL 7
000034  010766                  TRPSRV                  ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                  340                     ;SERVICE AT LEVEL 7
                000200          .=200
000200  000167  000574          JMP     START           ;GO TO START OF PROGRAM
                                        
                                        
                                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                        ;POINTERS TO SUBROUTINES CAN BE FOUND STARTING
                                        ;AT LOCATION "TRPTAB"
                                        
                104400                  SCOPE=TRAP+Y            ;SCOPE LOOP AND ITERATION HANDLER
                104401                  TYPE=TRAP+Y             ;TELETYPE OUTPUT ROUTINE
                104402                  OCTASC=TRAP+Y           ;OCTAL TO ASCII CONVERSION
                104403                  INSTR=TRAP+Y            ;INPUT ASCII STRING
                104404                  INSTER=TRAP+Y           ;STRING INPUT ERROR
                104405                  PARAM=TRAP+Y            ;CONVERT STRING TO OCTAL, CHECK LIMITS
                104406                  SAVO5P=TRAP+Y           ;SAVE R0-R5, PC
                104407                  RESO5=TRAP+Y            ;RESTORE R0-R5
                104410                  SCOPE1=TRAP+Y           ;CHECK FOR FREEZE ON CURRENT DATA
                000046          .=46
000046  010432                  LOGICAL
                000052          .=52
000052  040000                  40000
```

```
                        .=1000

                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

001000  012767  000340  176770  START:  MOV     #340,PS                 ;LOCK OUT INTERRUPTS
001006  012706  013540                  MOV     #STACK,SP               ;SET UP PROCESSOR STACK
001012  012737  012456  000024          MOV     #PFAIL,@#24             ;SET UP POWER FAIL TRAP
001020  005067  010670                  CLR     STFLG                   ;CLEAR TEST START FLAG
001024  005067  010624                  CLR     PASCNT                  ;CLEAR PASS COUNT
001030  005067  010622                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
001034  005067  010612                  CLR     ERRFLG                  ;CLEAR ERROR FLAG
001040  005067  010606                  CLR     ERRFLG                  ;CLEAR LAST ERROR PC
001044  104401  012622                  TYPE    ,MTITLE                 ;TYPE TITLE MESSAGE
001050  005767  010636                  TST     INIFLG                  ;CHECK INITIALIZATION FLAG
001054  001001                          BNE     VEC1                    ;IF NOT 0, CHECK SWITCHES
                                                                        ;FOR REINITIALIZATION
001056  000404                          BR      VEC2
001060  032767  000001  176502  VEC1:   BIT     #SW00,SWR               ;IF SW00=1, GET NEW VECTOR
001066  001445                          BEQ     BEGIN                   ;AND CSR
001070  012701  000300          VEC2:   MOV     #300,R1
001074  012702  000302                  MOV     #302,R2
001100  012703  000004                  MOV     #4,R3
001104  010211                  1$:     MOV     R2,(R1)                 ;RESTORE TRAPCATCHER
001106  005012                          CLR     (R2)                    ;IN FLOATING VECTOR AREA
001110  060301                          ADD     R3,R1
001112  060302                          ADD     R3,R2
001114  020127  001000                  CMP     R1,#1000
001120  001371                          BNE     1$
001122  104403                          INSTR                           ;INPUT ADDRESS OF DEVICE VECTOR
001124  012670                          MVECTOR                         ;MESSAGE "VECTOR ADDRESS-"
001126  104405                          PARAM                           ;CONVERT STRING TO OCTAL
001130  000300                          300                             ;LOW LIMIT
001132  000770                          770                             ;HIGH  LIMIT
001134  011642                          DHRVEC                          ;LOCATIONS TO BE FILLED
001136     003                  .BYTE   3                               ;NUMBER OF LOCATIONS
001137     004                  .BYTE   4                               ;LSB MASK
001140  104403                          INSTR                           ;INPUT ADDRESS OF DEVICE CSR
001142  012712                          MREGAD                          ;MESSAGE "CONTROL REGISTER ADDRESS-"
001144  104405                          PARAM                           ;CONVERT STRING TO OCTAL
001146  000000                          0                               ;LOW LIMIT
001150  177776                          177776                          ;HIGH LIMIT
001152  011620                          DHSCR                           ;LOCATIONS TO BE FILLED
001154     007                  .BYTE   7                               ;NUMBER OF LOCATIONS
001155     010                  .BYTE   10                              ;LSB MASK
001156  016767  010454  010454          MOV     DHSSR,DHSLR             ;SET UP ADDRESS OF SILO
001164  005267  010450                  INC     DHSLR                   ;STATUS REGISTER HIGH BYTE
001170  005767  010516                  TST     INIFLG                  ;IF INITIALIZATION FLAG
001174  001002                          BNE     BEGIN                   ;IS CLEARED
001176  005167  010510                  COM     INIFLG                  ;SET IT

                                        ;PROGRAM START
```

```
                                                ;CHECK FOR PROGRAM START AT SELECTED ADDRESS
                                                
001202  012767  000340  176566  BEGIN:  MOV     #340,PS         ;LOCK OUT INTERRUPTS
001210  012706  013540                  MOV     #STACK,SP       ;SET UP PROCESSOR STACK
001214  032767  000002  176346          BIT     #SW01,SWR       ;IF SW01=1
001222  001410                          BEQ     1$              ;GET PC FOR PROGRAM START
001224  104403                          INSTR                   ;GET PC
001226  013056                          MTSTPC                  ;MESSAGE "TEST PC"
001230  104405                          PARAM                   ;CONVERT STRING TO OCTAL
001232  000000                          0
001234  017500                          17500
001236  000207                          RETURN
001240     001                  .BYTE   1
001241     001                  .BYTE   1
001242  000410                          BR      2$
001244  012767  001274  010406  1$:     MOV     #T1,RETURN      ;NORMAL START, TEST 1
001252  005767  010436                  TST     STFLG           ;IF LOOPING, BYPASS TYPEOUT
001256  001004                          BNE     3$
001260  005167  010430                  COM     STFLG
001264  104401  013052          2$:     TYPE    ,MR             ;TYPE "R" TO INDICATE START
001270  000177  010364          3$:     JMP     @RETURN         ;START TESTING
```

```
   931
```

```
                                        ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
                                        ;ON LINE 0
                                        ;SET BREAK BIT FOR LINE 0
                                        ;TRANSMIT BINARY COUNT PATTERN ON LINE 0
                                        ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
                                        ;AND THAT IT IS A BREAK

001274  012767  000340  176474  T1:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
001302  012767  000020  010356        MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
001310  012767  001524  010344        MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
001316  012777  004000  010274        MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
001324  004767  010376                JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
                                                                      ;BUS ADDRESS MEMORY LOCATIONS
001330  012777  000000  010262        MOV     #0,@DHSCR               ;SELECT LINE 0
001336  012777  011722  010262        MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
001344  012777  177776  010256        MOV     #-2,@DHBC               ;TWO 0S WILL BE TRANSMITTED
001352  012777  033503  010244        MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
                                                                      ;CHARACTER LENGTH =8 BITS
001360  012777  000001  010244        MOV     #1,@DHBAR               ;SET BAR BIT FOR LINE 0
001366  122777  000002  010244  1$:   CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
001374  001374                       BNE     1$
001376  012777  004000  010214        MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
001404  012777  000000  010206        MOV     #0,@DHSCR               ;SELECT LINE 0
001412  012777  011754  010206        MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
001420  012777  177400  010202        MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
001426  012777  033503  010170        MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
001434  012777  000001  010172        MOV     #1,@DHBCR               ;SET BREAK BIT FOR LINE 0
001442  012777  000001  010162        MOV     #1,@DHBAR               ;SET BAR BIT FOR LINE 0
001450  005777  010156          2$:   TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
001454  001375                       BNE     2$                      ;TO BE TRANSMITTED
001456  122777  000001  010154        CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
001464  001407                       BEQ     3$                      ;1 CHARACTER WAS RECEIVED
001466  017704  010144                MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
001472  042704  000300                BIC     #300,R4                 ;CLEAR UNWANTED BITS
001476  012705  000400                MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
001502  104000                       HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
001504  017704  010112          3$:   MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
001510  026704  010702                CMP     RWRDO,R4                ;IS RECEIVED CHARACTER A BREAK
001514  001403                       BEQ     4$
001516  016705  010674                MOV     RWRDO,R5                ;(R5)=EXPECTED RECEIVED CHARACTER
001522  104001                       HLT     1                       ;RECEIVED DATA ERROR
001524  104400          4$:   SCOPE                                   ;CHECK FOR ITERATIONS, LOOP

                                        ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
                                        ;ON LINE 1
                                        ;SET BREAK BIT FOR LINE 1
                                        ;TRANSMIT BINARY COUNT PATTERN ON LINE 1
                                        ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
                                        ;AND THAT IT IS A BREAK

001526  012767  000340  176242  T2:   MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
001534  012767  000020  010124        MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
001542  012767  001756  010112        MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
001550  012777  004000  010042        MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
001556  004767  010144                JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
```

```
                                                        ;BUS ADDRESS MEMORY LOCATIONS
001562  012777  000001  010030          MOV     #1,@DHSCR       ;SELECT LINE 1
001570  012777  011722  010030          MOV     #NULL,@DHBA     ;SET UPT TO TRANSMIT 0 CHARACTER
001576  012777  177776  010024          MOV     #-2,@DHBC       ;TWO 0S WILL BE TRANSMITTED
001604  012777  033503  010012          MOV     #33503,@DHLPR   ;SET LINE SPEED=9600 BAUD
                                                        ;CHARACTER LENGTH =8 BITS
001612  012777  000002  010012          MOV     #2,@DHBAR       ;SET BAR BIT FOR LINE 1
001620  122777  000002  010012  1$:     CMPB    #2,@DHSLR       ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
001626  001374                          BNE     1$
001630  012777  004000  007762          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
001636  012777  000001  007754          MOV     #1,@DHSCR       ;SELECT LINE 1
001644  012777  011754  007754          MOV     #TBUF,@DHBA     ;SET UP TO TRANSMIT 400
001652  012777  177400  007750          MOV     #-400,@DHBC     ;(OCTAL) CHARACTERS
001660  012777  033503  007736          MOV     #33503,@DHLPR   ;LINE SPEED = 9600 BAUD
001666  012777  000002  007740          MOV     #2,@DHBCR       ;SET BREAK BIT FOR LINE 1
001674  012777  000002  007730          MOV     #2,@DHBAR       ;SET BAR BIT FOR LINE 1
001702  005777  007724          2$:     TST     @DHBAR          ;WAIT FOR ALL CHARACTERS
001706  001375                          BNE     2$              ;TO BE TRANSMITTED
001710  122777  000001  007722          CMPB    #1,@DHSLR       ;CHECK TO SEE THAT ONLY
001716  001407                          BEQ     3$              ;1 CHARACTER WAS RECEIVED
001720  017704  007712                  MOV     @DHSSR,R4       ;(R4)=ACTUAL RECEIVED DATA
001724  042704  000300                  BIC     #300,R4         ;CLEAR UNWANTED BITS
001730  012705  000400                  MOV     #400,R5         ;(R5)=EXPECTED SILO FILL LEVEL, 1
001734  104000                          HLT     0               ;MORE THAN ONE CHARACTER RECEIVED, ERROR
001736  017704  007660          3$:     MOV     @DHNRC,R4       ;READ NEXT RECEIVED CHARACTER REGISTER
001742  026704  010452                  CMP     RWRD1,R4                ;IS RECEIVED CHARACTER A BREAK
001746  001403                          BEQ     4$
001750  016705  010444                  MOV     RWRD1,R5        ;(R5)=EXPECTED RECEIVED CHARACTER
001754  104001                          HLT     1               ;RECEIVED DATA ERROR
001756  104400          4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP

                                ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
                                ;ON LINE 2
                                ;SET BREAK BIT FOR LINE 2
                                ;TRANSMIT BINARY COUNT PATTERN ON LINE 2
                                ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
                                ;AND THAT IT IS A BREAK

001760  012767  000340  176010  T3:     MOV     #340,PS         ;DISABLE ALL INTERRUPTS
001766  012767  000020  007672          MOV     #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
001774  012767  002210  007660          MOV     #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
002002  012777  004000  007610          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
002010  004767  007712                  JSR     PC,CLRALL       ;CLEAR ALL BUS ADDRESS AND
                                                        ;BUS ADDRESS MEMORY LOCATIONS
002014  012777  000002  007576          MOV     #2,@DHSCR       ;SELECT LINE 2
002022  012777  011722  007576          MOV     #NULL,@DHBA     ;SET UPT TO TRANSMIT 0 CHARACTER
002030  012777  177776  007572          MOV     #-2,@DHBC       ;TWO 0S WILL BE TRANSMITTED
002036  012777  033503  007560          MOV     #33503,@DHLPR   ;SET LINE SPEED=9600 BAUD
                                                        ;CHARACTER LENGTH =8 BITS
002044  012777  000004  007560          MOV     #4,@DHBAR       ;SET BAR BIT FOR LINE 2
002052  122777  000002  007560  1$:     CMPB    #2,@DHSLR       ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
002060  001374                          BNE     1$
002062  012777  004000  007530          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
002070  012777  000002  007522          MOV     #2,@DHSCR       ;SELECT LINE 2
002076  012777  011754  007522          MOV     #TBUF,@DHBA     ;SET UP TO TRANSMIT 400
002104  012777  177400  007516          MOV     #-400,@DHBC     ;(OCTAL) CHARACTERS
```

```
1044  002112  012777  033503  007504          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1045  002120  012777  000004  007506          MOV     #4,@DHBCR               ;SET BREAK BIT FOR LINE 2
1046  002126  012777  000004  007476          MOV     #4,@DHBAR               ;SET BAR BIT FOR LINE 2
1047  002134  005777  007472          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1048  002140  001375                          BNE     2$                      ;TO BE TRANSMITTED
1049  002142  122777  000001  007470          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1050  002150  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1051  002152  017704  007460                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1052  002156  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
1053  002162  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1054  002166  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1055  002170  017704  007426          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1056  002174  026704  010222                  CMP     RWRD2,R4                        ;IS RECEIVED CHARACTER A BREAK
1057  002200  001403                          BEQ     4$
1058  002202  016705  010214                  MOV     RWRD2,R5                ;(R5)=EXPECTED RECEIVED CHARACTER
1059  002206  104001                          HLT     1                       ;RECEIVED DATA ERROR
1060  002210  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1061
1062                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
1063                                          ;ON LINE 3
1064                                          ;SET BREAK BIT FOR LINE 3
1065                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 3
1066                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
1067                                          ;AND THAT IT IS A BREAK
1068
1069  002212  012767  000340  175556  T4:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1070  002220  012767  000020  007440          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1071  002226  012767  002442  007426          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1072  002234  012777  004000  007356          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1073  002242  004767  007460                  JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
1074                                                                          ;BUS ADDRESS MEMORY LOCATIONS
1075  002246  012777  000003  007344          MOV     #3,@DHSCR               ;SELECT LINE 3
1076  002254  012777  011722  007344          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1077  002262  012777  177776  007340          MOV     #-2,@DHBC               ;TWO 0S WILL BE TRANSMITTED
1078  002270  012777  033503  007326          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1079                                                                          ;CHARACTER LENGTH =8 BITS
1080  002276  012777  000010  007326          MOV     #10,@DHBAR              ;SET BAR BIT FOR LINE 3
1081  002304  122777  000002  007326  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1082  002312  001374                          BNE     1$
1083  002314  012777  004000  007276          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1084  002322  012777  000003  007270          MOV     #3,@DHSCR               ;SELECT LINE 3
1085  002330  012777  011754  007270          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1086  002336  012777  177400  007264          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1087  002344  012777  033503  007252          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1088  002352  012777  000010  007254          MOV     #10,@DHBCR              ;SET BREAK BIT FOR LINE 3
1089  002360  012777  000010  007244          MOV     #10,@DHBAR              ;SET BAR BIT FOR LINE 3
1090  002366  005777  007240          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1091  002372  001375                          BNE     2$                      ;TO BE TRANSMITTED
1092  002374  122777  000001  007236          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1093  002402  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1094  002404  017704  007226                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1095  002410  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
1096  002414  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1097  002420  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1098  002422  017704  007174          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1099  002426  026704  007772                  CMP     RWRD3,R4                        ;IS RECEIVED CHARACTER A BREAK
```

```
 1100  002432  001403                          BEQ     4$
 1101  002434  016705  007764                  MOV     RWRD3,R5                ;(R5)=EXPECTED RECEIVED CHARACTER
 1102  002440  104001                          HLT     1                       ;RECEIVED DATA ERROR
 1103  002442  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
 1104
 1105                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
 1106                                          ;ON LINE 4
 1107                                          ;SET BREAK BIT FOR LINE 4
 1108                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 4
 1109                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
 1110                                          ;AND THAT IT IS A BREAK
 1111
 1112  002444  012767  000340  175324  T5:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
 1113  002452  012767  000020  007206          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
 1114  002460  012767  002674  007174          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
 1115  002466  012777  004000  007124          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1116  002474  004767  007226                  JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
 1117                                                                          ;BUS ADDRESS MEMORY LOCATIONS
 1118  002500  012777  000004  007112          MOV     #4,@DHSCR               ;SELECT LINE 4
 1119  002506  012777  011722  007112          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
 1120  002514  012777  177776  007106          MOV     #-2,@DHBC               ;TWO 0S WILL BE TRANSMITTED
 1121  002522  012777  033503  007074          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
 1122                                                                          ;CHARACTER LENGTH =8 BITS
 1123  002530  012777  000020  007074          MOV     #20,@DHBAR              ;SET BAR BIT FOR LINE 4
 1124  002536  122777  000002  007074  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
 1125  002544  001374                          BNE     1$
 1126  002546  012777  004000  007044          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1127  002554  012777  000004  007036          MOV     #4,@DHSCR               ;SELECT LINE 4
 1128  002562  012777  011754  007036          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
 1129  002570  012777  177400  007032          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
 1130  002576  012777  033503  007020          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
 1131  002604  012777  000020  007022          MOV     #20,@DHBCR              ;SET BREAK BIT FOR LINE 4
 1132  002612  012777  000020  007012          MOV     #20,@DHBAR              ;SET BAR BIT FOR LINE 4
 1133  002620  005777  007006          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
 1134  002624  001375                          BNE     2$                      ;TO BE TRANSMITTED
 1135  002626  122777  000001  007004          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
 1136  002634  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
 1137  002636  017704  006774                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
 1138  002642  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
 1139  002646  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
 1140  002652  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
 1141  002654  017704  006742          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
 1142  002660  026704  007542                  CMP     RWRD4,R4                        ;IS RECEIVED CHARACTER A BREAK
 1143  002664  001403                          BEQ     4$
 1144  002666  016705  007534                  MOV     RWRD4,R5                ;(R5)=EXPECTED RECEIVED CHARACTER
 1145  002672  104001                          HLT     1                       ;RECEIVED DATA ERROR
 1146  002674  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
 1147
 1148                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
 1149                                          ;ON LINE 5
 1150                                          ;SET BREAK BIT FOR LINE 5
 1151                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 5
 1152                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
 1153                                          ;AND THAT IT IS A BREAK
 1154
 1155  002676  012767  000340  175072  T6:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
```

```
1156  002704  012767  000020  006754          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1157  002712  012767  003126  006742          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1158  002720  012777  004000  006672          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1159  002726  004767  006774                  JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
1160                                                                          ;BUS ADDRESS MEMORY LOCATIONS
1161  002732  012777  000005  006660          MOV     #5,@DHSCR               ;SELECT LINE 5
1162  002740  012777  011722  006660          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1163  002746  012777  177776  006654          MOV     #-2,@DHBC               ;TWO 0S WILL BE TRANSMITTED
1164  002754  012777  033503  006642          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1165                                                                          ;CHARACTER LENGTH =8 BITS
1166  002762  012777  000040  006642          MOV     #40,@DHBAR              ;SET BAR BIT FOR LINE 5
1167  002770  122777  000002  006642  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1168  002776  001374                          BNE     1$
1169  003000  012777  004000  006612          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1170  003006  012777  000005  006604          MOV     #5,@DHSCR               ;SELECT LINE 5
1171  003014  012777  011754  006604          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1172  003022  012777  177400  006600          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1173  003030  012777  033503  006566          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1174  003036  012777  000040  006570          MOV     #40,@DHBCR              ;SET BREAK BIT FOR LINE 5
1175  003044  012777  000040  006560          MOV     #40,@DHBAR              ;SET BAR BIT FOR LINE 5
1176  003052  005777  006554          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1177  003056  001375                          BNE     2$                      ;TO BE TRANSMITTED
1178  003060  122777  000001  006552          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1179  003066  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1180  003070  017704  006542                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1181  003074  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
1182  003100  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1183  003104  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1184  003106  017704  006510          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1185  003112  026704  007312                  CMP     RWRD5,R4                        ;IS RECEIVED CHARACTER A BREAK
1186  003116  001403                          BEQ     4$
1187  003120  016705  007304                  MOV     RWRD5,R5                ;(R5)=EXPECTED RECEIVED CHARACTER
1188  003124  104001                          HLT     1                       ;RECEIVED DATA ERROR
1189  003126  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1190
1191                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
1192                                          ;ON LINE 6
1193                                          ;SET BREAK BIT FOR LINE 6
1194                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 6
1195                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
1196                                          ;AND THAT IT IS A BREAK
1197
1198  003130  012767  000340  174640  T7:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1199  003136  012767  000020  006522          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1200  003144  012767  003360  006510          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1201  003152  012777  004000  006440          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1202  003160  004767  006542                  JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
1203                                                                          ;BUS ADDRESS MEMORY LOCATIONS
1204  003164  012777  000006  006426          MOV     #6,@DHSCR               ;SELECT LINE 6
1205  003172  012777  011722  006426          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1206  003200  012777  177776  006422          MOV     #-2,@DHBC               ;TWO 0S WILL BE TRANSMITTED
1207  003206  012777  033503  006410          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1208                                                                          ;CHARACTER LENGTH =8 BITS
1209  003214  012777  000100  006410          MOV     #100,@DHBAR             ;SET BAR BIT FOR LINE 6
1210  003222  122777  000002  006410  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1211  003230  001374                          BNE     1$
```

```
 1212  003232  012777  004000  006360          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1213  003240  012777  000006  006352          MOV     #6,@DHSCR               ;SELECT LINE 6
 1214  003246  012777  011754  006352          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
 1215  003254  012777  177400  006346          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
 1216  003262  012777  033503  006334          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
 1217  003270  012777  000100  006336          MOV     #100,@DHBCR             ;SET BREAK BIT FOR LINE 6
 1218  003276  012777  000100  006326          MOV     #100,@DHBAR             ;SET BAR BIT FOR LINE 6
 1219  003304  005777  006322          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
 1220  003310  001375                          BNE     2$                      ;TO BE TRANSMITTED
 1221  003312  122777  000001  006320          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
 1222  003320  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
 1223  003322  017704  006310                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
 1224  003326  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
 1225  003332  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
 1226  003336  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
 1227  003340  017704  006256          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
 1228  003344  026704  007062                  CMP     RWRD6,R4                ;IS RECEIVED CHARACTER A BREAK
 1229  003350  001403                          BEQ     4$
 1230  003352  016705  007054                  MOV     RWRD6,R5                ;(R5)=EXPECTED RECEIVED CHAPACTER
 1231  003356  104001                          HLT     1                       ;RECEIVED DATA ERROR
 1232  003360  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
 1233
 1234                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
 1235                                          ;ON LINE 7
 1236                                          ;SET BREAK BIT FOR LINE 7
 1237                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 7
 1238                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
 1239                                          ;AND THAT IT IS A BREAK
 1240
 1241  003362  012767  000340  174406  T10:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
 1242  003370  012767  000020  006270          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
 1243  003376  012767  003612  006256          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
 1244  003404  012777  004000  006206          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1245  003412  004767  006310                  JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
 1246                                                                          ;BUS ADDRESS MEMORY LOCATIONS
 1247  003416  012777  000007  006174          MOV     #7,@DHSCR               ;SELECT LINE 7
 1248  003424  012777  011722  006174          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
 1249  003432  012777  177776  006170          MOV     #-2,@DHBC               ;TWO 0S WILL BE TRANSMITTED
 1250  003440  012777  033503  006156          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
 1251                                                                          ;CHARACTER LENGTH =8 BITS
 1252  003446  012777  000200  006156          MOV     #200,@DHBAR             ;SET BAR BIT FOR LINE 7
 1253  003454  122777  000002  006156  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
 1254  003462  001374                          BNE     1$
 1255  003464  012777  004000  006126          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1256  003472  012777  000007  006120          MOV     #7,@DHSCR               ;SELECT LINE 7
 1257  003500  012777  011754  006120          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
 1258  003506  012777  177400  006114          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
 1259  003514  012777  033503  006102          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
 1260  003522  012777  000200  006104          MOV     #200,@DHBCR             ;SET BREAK BIT FOR LINE 7
 1261  003530  012777  000200  006074          MOV     #200,@DHBAR             ;SET BAR BIT FOR LINE 7
 1262  003536  005777  006070          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
 1263  003542  001375                          BNE     2$                      ;TO BE TRANSMITTED
 1264  003544  122777  000001  006066          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
 1265  003552  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
 1266  003554  017704  006056                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
 1267  003560  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
```

```
1268  003564  012705  000400              MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1269  003570  104000                      HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1270  003572  017704  006024      3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1271  003576  026704  006632              CMP     RWRD7,R4                        ;IS RECEIVED CHARACTER A BREAK
1272  003602  001403                      BEQ     4$
1273  003604  016705  006624              MOV     RWRD7,R5                ;(R5)=EXPECTED RECEIVED CHARACTER
1274  003610  104001                      HLT     1                       ;RECEIVED DATA ERROR
1275  003612  104400              4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1276
1277                                      ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
1278                                      ;ON LINE 10
1279                                      ;SET BREAK BIT FOR LINE 10
1280                                      ;TRANSMIT BINARY COUNT PATTERN ON LINE 10
1281                                      ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
1282                                      ;AND THAT IT IS A BREAK
1283
1284  003614  012767  000340  174154  T11:  MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1285  003622  012767  000020  006036      MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1286  003630  012767  004044  006024      MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1287  003636  012777  004000  005754      MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1288  003644  004767  006056              JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
1289                                                                      ;BUS ADDRESS MEMORY LOCATIONS
1290  003650  012777  000010  005742      MOV     #10,@DHSCR              ;SELECT LINE 10
1291  003656  012777  011722  005742      MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1292  003664  012777  177776  005736      MOV     #-2,@DHBC               ;TWO OS WILL BE TRANSMITTED
1293  003672  012777  033503  005724      MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1294                                                                      ;CHARACTER LENGTH =8 BITS
1295  003700  012777  000400  005724      MOV     #400,@DHBAR             ;SET BAR BIT FOR LINE 10
1296  003706  122777  000002  005724  1$:  CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1297  003714  001374                      BNE     1$
1298  003716  012777  004000  005674      MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1299  003724  012777  000010  005666      MOV     #10,@DHSCR              ;SELECT LINE 10
1300  003732  012777  011754  005666      MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1301  003740  012777  177400  005662      MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1302  003746  012777  033503  005650      MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1303  003754  012777  000400  005652      MOV     #400,@DHBCR             ;SET BREAK BIT FOR LINE 10
1304  003762  012777  000400  005642      MOV     #400,@DHBAR             ;SET BAR BIT FOR LINE 10
1305  003770  005777  005636          2$: TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1306  003774  001375                      BNE     2$                      ;TO BE TRANSMITTED
1307  003776  122777  000001  005634      CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1308  004004  001407                      BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1309  004006  017704  005624              MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1310  004012  042704  000300              BIC     #300,R4                 ;CLEAR UNWANTED BITS
1311  004016  012705  000400              MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1312  004022  104000                      HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1313  004024  017704  005572          3$: MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1314  004030  026704  006402              CMP     RWRD10,R4                       ;IS RECEIVED CHARACTER A BREAK
1315  004034  001403                      BEQ     4$
1316  004036  016705  006374              MOV     RWRD10,R5               ;(R5)=EXPECTED RECEIVED CHARACTER
1317  004042  104001                      HLT     1                       ;RECEIVED DATA ERROR
1318  004044  104400              4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1319
1320                                      ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
1321                                      ;ON LINE 11
1322                                      ;SET BREAK BIT FOR LINE 11
1323                                      ;TRANSMIT BINARY COUNT PATTERN ON LINE 11
```

```
                                                ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
                                                ;AND THAT IT IS A BREAK
                                                
004046  012767  000340  173722  T12:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
004054  012767  000020  005604        MOV   #20.,ICOUNT      ;SET UP FOR 20 ITERATIONS
004062  012767  004276  005572        MOV   #4$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
004070  012777  004000  005522        MOV   #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
004076  004767  005624                JSR   PC,CLRALL        ;CLEAR ALL BUS ADDRESS AND
                                                ;BUS ADDRESS MEMORY LOCATIONS
004102  012777  000011  005510        MOV   #11,@DHSCR       ;SELECT LINE 11
004110  012777  011722  005510        MOV   #NULL,@DHBA      ;SET UPT TO TRANSMIT 0 CHARACTER
004116  012777  177776  005504        MOV   #-2,@DHBC        ;TWO 0S WILL BE TRANSMITTED
004124  012777  033503  005472        MOV   #33503,@DHLPR    ;SET LINE SPEED=9600 BAUD
                                                ;CHARACTER LENGTH =8 BITS
004132  012777  001000  005472        MOV   #1000,@DHBAR     ;SET BAR BIT FOR LINE 11
004140  122777  000002  005472  1$:   CMPB  #2,@DHSLR        ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
004146  001374                       BNE   1$
004150  012777  004000  005442        MOV   #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
004156  012777  000011  005434        MOV   #11,@DHSCR       ;SELECT LINE 11
004164  012777  011754  005434        MOV   #TBUF,@DHBA      ;SET UP TO TRANSMIT 400
004172  012777  177400  005430        MOV   #-400,@DHBC      ;(OCTAL) CHARACTERS
004200  012777  033503  005416        MOV   #33503,@DHLPR    ;LINE SPEED = 9600 BAUD
004206  012777  001000  005420        MOV   #1000,@DHBCR     ;SET BREAK BIT FOR LINE 11
004214  012777  001000  005410        MOV   #1000,@DHBAR     ;SET BAR BIT FOR LINE 11
004222  005777  005404          2$:   TST   @DHBAR           ;WAIT FOR ALL CHARACTERS
004226  001375                       BNE   2$               ;TO BE TRANSMITTED
004230  122777  000001  005402        CMPB  #1,@DHSLR        ;CHECK TO SEE THAT ONLY
004236  001407                       BEQ   3$               ;1 CHARACTER WAS RECEIVED
004240  017704  005372                MOV   @DHSSR,R4        ;(R4)=ACTUAL RECEIVED DATA
004244  042704  000300                BIC   #300,R4          ;CLEAR UNWANTED BITS
004250  012705  000400                MOV   #400,R5          ;(R5)=EXPECTED SILO FILL LEVEL, 1
004254  104000                       HLT   0                ;MORE THAN ONE CHARACTER RECEIVED, ERROR
004256  017704  005340          3$:   MOV   @DHNRC,R4        ;READ NEXT RECEIVED CHARACTER REGISTER
004262  026704  006152                CMP   RWRD11,R4                ;IS RECEIVED CHARACTER A BREAK
004266  001403                       BEQ   4$
004270  016705  006144                MOV   RWRD11,R5        ;(R5)=EXPECTED RECEIVED CHARACTER
004274  104001                       HLT   1                ;RECEIVED DATA ERROR
004276  104400                4$:    SCOPE                  ;CHECK FOR ITERATIONS, LOOP
                                                
                                                ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
                                                ;ON LINE 12
                                                ;SET BREAK BIT FOR LINE 12
                                                ;TRANSMIT BINARY COUNT PATTERN ON LINE 12
                                                ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
                                                ;AND THAT IT IS A BREAK
                                                
004300  012767  000340  173470  T13:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
004306  012767  000020  005352        MOV   #20,ICOUNT       ;SET UP FOR 20 ITERATIONS
004314  012767  004530  005340        MOV   #4$,ESCAPE       ;SET UP TO ESCAPE TO NEXT TEST
004322  012777  004000  005270        MOV   #BIT11,@DHSCR    ;MASTER CLEAR INTERFACE
004330  004767  005372                JSR   PC,CLRALL        ;CLEAR ALL BUS ADDRESS AND
                                                ;BUS ADDRESS MEMORY LOCATIONS
004334  012777  000012  005256        MOV   #12,@DHSCR       ;SELECT LINE 12
004342  012777  011722  005256        MOV   #NULL,@DHBA      ;SET UPT TO TRANSMIT 0 CHARACTER
004350  012777  177776  005252        MOV   #-2,@DHBC        ;TWO 0S WILL BE TRANSMITTED
004356  012777  033503  005240        MOV   #33503,@DHLPR    ;SET LINE SPEED=9600 BAUD
```

```
1380
1381  004364  012777  002000  005240          MOV     #2000,@DHBAR            ;SET BAR BIT FOR LINE 12
1382  004372  122777  000002  005240  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1383  004400  001374                          BNE     1$
1384  004402  012777  004000  005210          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1385  004410  012777  000012  005202          MOV     #12,@DHSCR              ;SELECT LINE 12
1386  004416  012777  011754  005202          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1387  004424  012777  177400  005176          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1388  004432  012777  033503  005164          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1389  004440  012777  002000  005166          MOV     #2000,@DHBCR            ;SET BREAK BIT FOR LINE 12
1390  004446  012777  002000  005156          MOV     #2000,@DHBAR            ;SET BAR BIT FOR LINE 12
1391  004454  005777  005152          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1392  004460  001375                          BNE     2$                      ;TO BE TRANSMITTED
1393  004462  122777  000001  005150          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1394  004470  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1395  004472  017704  005140                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1396  004476  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
1397  004502  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1398  004506  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1399  004510  017704  005106          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1400  004514  026704  005722                  CMP     RWRD12,R4                       ;IS RECEIVED CHARACTER A BREAK
1401  004520  001403                          BEQ     4$
1402  004522  016705  005714                  MOV     RWRD12,R5               ;(R5)=EXPECTED RECEIVED CHARACTER
1403  004526  104001                          HLT     1                       ;RECEIVED DATA ERROR
1404  004530  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1405
1406                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
1407                                          ;ON LINE 13
1408                                          ;SET BREAK BIT FOR LINE 13
1409                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 13
1410                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
1411                                          ;AND THAT IT IS A BREAK
1412
1413  004532  012767  000340  173236  T14:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1414  004540  012767  000020  005120          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1415  004546  012767  004762  005106          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1416  004554  012777  004000  005036          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1417  004562  004767  005140                  JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
1418                                                                          ;BUS ADDRESS MEMORY LOCATIONS
1419  004566  012777  000013  005024          MOV     #13,@DHSCR              ;SELECT LINE 13
1420  004574  012777  011722  005024          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1421  004602  012777  177776  005020          MOV     #-2,@DHBC               ;TWO OS WILL BE TRANSMITTED
1422  004610  012777  033503  005006          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1423                                                                          ;CHARACTER LENGTH =8 BITS
1424  004616  012777  004000  005006          MOV     #4000,@DHBAR            ;SET BAR BIT FOR LINE 13
1425  004624  122777  000002  005006  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1426  004632  001374                          BNE     1$
1427  004634  012777  004000  004756          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1428  004642  012777  000013  004750          MOV     #13,@DHSCR              ;SELECT LINE 13
1429  004650  012777  011754  004750          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1430  004656  012777  177400  004744          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1431  004664  012777  033503  004732          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1432  004672  012777  004000  004734          MOV     #4000,@DHBCR            ;SET BREAK BIT FOR LINE 13
1433  004700  012777  004000  004724          MOV     #4000,@DHBAR            ;SET BAR BIT FOR LINE 13
1434  004706  005777  004720          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1435  004712  001375                          BNE     2$                      ;TO BE TRANSMITTED
```

```
1436  004714  122777  000001  004716          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1437  004722  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1438  004724  017704  004706                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1439  004730  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
1440  004734  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1441  004740  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1442  004742  017704  004654          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1443  004746  026704  005472                  CMP     RWRD13,R4                       ;IS RECEIVED CHARACTER A BREAK
1444  004752  001403                          BEQ     4$
1445  004754  016705  005464                  MOV     RWRD13,R5               ;(R5)=EXPECTED RECEIVED CHARACTER
1446  004760  104001                          HLT     1                       ;RECEIVED DATA ERROR
1447  004762  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1448
1449                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
1450                                          ;ON LINE 14
1451                                          ;SET BREAK BIT FOR LINE 14
1452                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 14
1453                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
1454                                          ;AND THAT IT IS A BREAK
1455
1456  004764  012767  000340  173004  T15:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1457  004772  012767  000020  004666          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1458  005000  012767  005214  004654          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1459  005006  012777  004000  004604          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1460  005014  004767  004706                  JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
1461                                                                          ;BUS ADDRESS MEMORY LOCATIONS
1462  005020  012777  000014  004572          MOV     #14,@DHSCR              ;SELECT LINE 14
1463  005026  012777  011722  004572          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1464  005034  012777  177776  004566          MOV     #-2,@DHBC               ;TWO 0S WILL BE TRANSMITTED
1465  005042  012777  033503  004554          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1466                                                                          ;CHARACTER LENGTH =8 BITS
1467  005050  012777  010000  004554          MOV     #10000,@DHBAR           ;SET BAR BIT FOR LINE 14
1468  005056  122777  000002  004554  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1469  005064  001374                          BNE     1$
1470  005066  012777  004000  004524          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1471  005074  012777  000014  004516          MOV     #14,@DHSCR              ;SELECT LINE 14
1472  005102  012777  011754  004516          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1473  005110  012777  177400  004512          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1474  005116  012777  033503  004500          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1475  005124  012777  010000  004502          MOV     #10000,@DHBCR           ;SET BREAK BIT FOR LINE 14
1476  005132  012777  010000  004472          MOV     #10000,@DHBAR           ;SET BAR BIT FOR LINE 14
1477  005140  005777  004466          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1478  005144  001375                          BNE     2$                      ;TO BE TRANSMITTED
1479  005146  122777  000001  004464          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1480  005154  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1481  005156  017704  004454                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1482  005162  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
1483  005166  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1484  005172  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1485  005174  017704  004422          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1486  005200  026704  005242                  CMP     RWRD14,R4                       ;IS RECEIVED CHARACTER A BREAK
1487  005204  001403                          BEQ     4$
1488  005206  016705  005234                  MOV     RWRD14,R5               ;(R5)=EXPECTED RECEIVED CHARACTER
1489  005212  104001                          HLT     1                       ;RECEIVED DATA ERROR
1490  005214  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1491
```

```
                                        ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
                                        ;ON LINE 15
                                        ;SET BREAK BIT FOR LINE 15
                                        ;TRANSMIT BINARY COUNT PATTERN ON LINE 15
                                        ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
                                        ;AND THAT IT IS A BREAK

005216  012767  000340  172552  T16:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
005224  012767  000020  004434        MOV   #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
005232  012767  005446  004422        MOV   #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
005240  012777  004000  004352        MOV   #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
005246  004767  004454              JSR   PC,CLRALL       ;CLEAR ALL BUS ADDRESS AND
                                                          ;BUS ADDRESS MEMORY LOCATIONS
005252  012777  000015  004340        MOV   #15,@DHSCR      ;SELECT LINE 15
005260  012777  011722  004340        MOV   #NULL,@DHBA     ;SET UPT TO TRANSMIT 0 CHARACTER
005266  012777  177776  004334        MOV   #-2,@DHBC       ;TWO 0S WILL BE TRANSMITTED
005274  012777  033503  004322        MOV   #33503,@DHLPR   ;SET LINE SPEED=9600 BAUD
                                                          ;CHARACTER LENGTH =8 BITS
005302  012777  020000  004322        MOV   #20000,@DHBAR   ;SET BAR BIT FOR LINE 15
005310  122777  000002  004322  1$:   CMPB  #2,@DHSLR       ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
005316  001374                      BNE   1$
005320  012777  004000  004272        MOV   #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
005326  012777  000015  004264        MOV   #15,@DHSCR      ;SELECT LINE 15
005334  012777  011754  004264        MOV   #TBUF,@DHBA     ;SET UP TO TRANSMIT 400
005342  012777  177400  004260        MOV   #-400,@DHBC     ;(OCTAL) CHARACTERS
005350  012777  033503  004246        MOV   #33503,@DHLPR   ;LINE SPEED = 9600 BAUD
005356  012777  020000  004250        MOV   #20000,@DHBCR   ;SET BREAK BIT FOR LINE 15
005364  012777  020000  004240        MOV   #20000,@DHBAR   ;SET BAR BIT FOR LINE 15
005372  005777  004234        2$:   TST   @DHBAR          ;WAIT FOR ALL CHARACTERS
005376  001375                      BNE   2$              ;TO BE TRANSMITTED
005400  122777  000001  004232        CMPB  #1,@DHSLR       ;CHECK TO SEE THAT ONLY
005406  001407                      BEQ   3$              ;1 CHARACTER WAS RECEIVED
005410  017704  004222              MOV   @DHSSR,R4       ;(R4)=ACTUAL RECEIVED DATA
005414  042704  000300              BIC   #300,R4         ;CLEAR UNWANTED BITS
005420  012705  000400              MOV   #400,R5         ;(R5)=EXPECTED SILO FILL LEVEL, 1
005424  104000                      HLT   0               ;MORE THAN ONE CHARACTER RECEIVED, ERROR
005426  017704  004170        3$:   MOV   @DHNRC,R4       ;READ NEXT RECEIVED CHARACTER REGISTER
005432  026704  005012              CMP   RWRD15,R4               ;IS RECEIVED CHARACTER A BREAK
005436  001403                      BEQ   4$
005440  016705  005004              MOV   RWRD15,R5       ;(R5)=EXPECTED RECEIVED CHARACTER
005444  104001                      HLT   1               ;RECEIVED DATA ERROR
005446  104400              4$:   SCOPE                   ;CHECK FOR ITERATIONS, LOOP

                                        ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
                                        ;ON LINE 16
                                        ;SET BREAK BIT FOR LINE 16
                                        ;TRANSMIT BINARY COUNT PATTERN ON LINE 16
                                        ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
                                        ;AND THAT IT IS A BREAK

005450  012767  000340  172320  T17:  MOV   #340,PS          ;DISABLE ALL INTERRUPTS
005456  012767  000020  004202        MOV   #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
005464  012767  005700  004170        MOV   #4$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
005472  012777  004000  004120        MOV   #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
005500  004767  004222              JSR   PC,CLRALL       ;CLEAR ALL BUS ADDRESS AND
                                                          ;BUS ADDRESS MEMORY LOCATIONS
```

```
1548  005504  012777  000016  004106          MOV     #16,@DHSCR              ;SELECT LINE 16
1549  005512  012777  011722  004106          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1550  005520  012777  177776  004102          MOV     #-2,@DHBC               ;TWO OS WILL BE TRANSMITTED
1551  005526  012777  033503  004070          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1552                                                                          ;CHARACTER LENGTH =8 BITS
1553  005534  012777  040000  004070          MOV     #40000,@DHBAR           ;SET BAR BIT FOR LINE 16
1554  005542  122777  000002  004070  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1555  005550  001374                         BNE     1$
1556  005552  012777  004000  004040          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1557  005560  012777  000016  004032          MOV     #16,@DHSCR              ;SELECT LINE 16
1558  005566  012777  011754  004032          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1559  005574  012777  177400  004026          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1560  005602  012777  033503  004014          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
1561  005610  012777  040000  004016          MOV     #40000,@DHBCR           ;SET BREAK BIT FOR LINE 16
1562  005616  012777  040000  004006          MOV     #40000,@DHBAR           ;SET BAR BIT FOR LINE 16
1563  005624  005777  004002          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1564  005630  001375                         BNE     2$                      ;TO BE TRANSMITTED
1565  005632  122777  000001  004000          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1566  005640  001407                         BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1567  005642  017704  003770                 MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1568  005646  042704  000300                 BIC     #300,R4                 ;CLEAR UNWANTED BITS
1569  005652  012705  000400                 MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1570  005656  104000                         HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1571  005660  017704  003736          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1572  005664  026704  004562                 CMP     RWRD16,R4                       ;IS RECEIVED CHARACTER A BREAK
1573  005670  001403                         BEQ     4$
1574  005672  016705  004554                 MOV     RWRD16,R5               ;(R5)=EXPECTED RECEIVED CHARACTER
1575  005676  104001                         HLT     1                       ;RECEIVED DATA ERROR
1576  005700  104400                  4$:    SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1577
1578                                          ;FLUSH UART BY TRANSMITTING 2 NULL CHARACTERS
1579                                          ;ON LINE 17
1580                                          ;SET BREAK BIT FOR LINE 17
1581                                          ;TRANSMIT BINARY COUNT PATTERN ON LINE 17
1582                                          ;VERIFY HTAT ONLY 1 CHARACTER IS RECEIVED
1583                                          ;AND THAT IT IS A BREAK
1584
1585  005702  012767  000340  172066  T20:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1586  005710  012767  000020  003750          MOV     #20.,ICOUNT             ;SET UP FOR 20 ITERATIONS
1587  005716  012767  006132  003736          MOV     #4$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1588  005724  012777  004000  003666          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1589  005732  004767  003770                 JSR     PC,CLRALL               ;CLEAR ALL BUS ADDRESS AND
1590                                                                          ;BUS ADDRESS MEMORY LOCATIONS
1591  005736  012777  000017  003654          MOV     #17,@DHSCR              ;SELECT LINE 17
1592  005744  012777  011722  003654          MOV     #NULL,@DHBA             ;SET UPT TO TRANSMIT 0 CHARACTER
1593  005752  012777  177776  003650          MOV     #-2,@DHBC               ;TWO OS WILL BE TRANSMITTED
1594  005760  012777  033503  003636          MOV     #33503,@DHLPR           ;SET LINE SPEED=9600 BAUD
1595                                                                          ;CHARACTER LENGTH =8 BITS
1596  005766  012777  100000  003636          MOV     #100000,@DHBAR          ;SET BAR BIT FOR LINE 17
1597  005774  122777  000002  003636  1$:     CMPB    #2,@DHSLR               ;WAIT FOR 2 CHARACTERS TO BE RECEIVED
1598  006002  001374                         BNE     1$
1599  006004  012777  004000  003606          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1600  006012  012777  000017  003600          MOV     #17,@DHSCR              ;SELECT LINE 17
1601  006020  012777  011754  003600          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT 400
1602  006026  012777  177400  003574          MOV     #-400,@DHBC             ;(OCTAL) CHARACTERS
1603  006034  012777  033503  003562          MOV     #33503,@DHLPR           ;LINE SPEED = 9600 BAUD
```

```
1604  006042  012777  100000  003564          MOV     #100000,@DHBCR          ;SET BREAK BIT FOR LINE 17
1605  006050  012777  100000  003554          MOV     #100000,@DHBAR          ;SET BAR BIT FOR LINE 17
1606  006056  005777  003550          2$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS
1607  006062  001375                          BNE     2$                      ;TO BE TRANSMITTED
1608  006064  122777  000001  003546          CMPB    #1,@DHSLR               ;CHECK TO SEE THAT ONLY
1609  006072  001407                          BEQ     3$                      ;1 CHARACTER WAS RECEIVED
1610  006074  017704  003536                  MOV     @DHSSR,R4               ;(R4)=ACTUAL RECEIVED DATA
1611  006100  042704  000300                  BIC     #300,R4                 ;CLEAR UNWANTED BITS
1612  006104  012705  000400                  MOV     #400,R5                 ;(R5)=EXPECTED SILO FILL LEVEL, 1
1613  006110  104000                          HLT     0                       ;MORE THAN ONE CHARACTER RECEIVED, ERROR
1614  006112  017704  003504          3$:     MOV     @DHNRC,R4               ;READ NEXT RECEIVED CHARACTER REGISTER
1615  006116  026704  004332                  CMP     RWRD17,R4                       ;IS RECEIVED CHARACTER A BREAK
1616  006122  001403                          BEQ     4$
1617  006124  016705  004324                  MOV     RWRD17,R5               ;(R5)=EXPECTED RECEIVED CHARACTER
1618  006130  104001                          HLT     1                       ;RECEIVED DATA ERROR
1619  006132  104400                  4$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1620
1621                                          ;SET HALF DUPLEX ON LINE 0
1622                                          ;TRANSMITI A BINARY COUNT PATTERN
1623                                          ;VERIFY THAT NO CHARACTERS ARE RECEIVED
1624
1625  006134  012767  000340  171634  T21:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1626  006142  012767  000020  003516          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1627  006150  012767  006244  003504          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1628  006156  012777  004000  003434          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1629  006164  004767  003536                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
1630                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
1631  006170  012777  000000  003422          MOV     #0,@DHSCR               ;SELECT LINE 0
1632  006176  012777  011754  003422          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
1633  006204  012777  177400  003416          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
1634  006212  012777  073503  003404          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
1635                                                                          ;LINE SPEED =9600 BAUD
1636                                                                          ;CHARACTER LENGTH = 8 BITS
1637  006220  012777  000001  003404          MOV     #1,@DHBAR               ;SET BAR BIT FOR LINE 0
1638  006226  005777  003400          1$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
1639  006232  001375                          BNE     1$
1640  006234  105777  003360                  TSTB    @DHSCR                  ;WERE ANY CHARACTERS RECEIVED
1641  006240  100001                          BPL     2$
1642  006242  104002                          HLT     2                       ;RECEIVER NOT BLINDED, ERROR
1643  006244  104400                  2$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
1644
1645                                          ;SET HALF DUPLEX ON LINE 1
1646                                          ;TRANSMITI A BINARY COUNT PATTERN
1647                                          ;VERIFY THAT NO CHARACTERS ARE RECEIVED
1648
1649  006246  012767  000340  171522  T22:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
1650  006254  012767  000020  003404          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
1651  006262  012767  006356  003372          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
1652  006270  012777  004000  003322          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
1653  006276  004767  003424                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
1654                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
1655  006302  012777  000001  003310          MOV     #1,@DHSCR               ;SELECT LINE 1
1656  006310  012777  011754  003310          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
1657  006316  012777  177400  003304          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
1658  006324  012777  073503  003272          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
1659                                                                          ;LINE SPEED =9600 BAUD
```

```
                                                        ;CHARACTER LENGTH = 8 BITS
006332  012777  000002  003272          MOV     #2,@DHBAR       ;SET BAR BIT FOR LINE 1
006340  005777  003266          1$:     TST     @DHBAR          ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
006344  001375                          BNE     1$
006346  105777  003246                  TSTB    @DHSCR          ;WERE ANY CHARACTERS RECEIVED
006352  100001                          BPL     2$
006354  104002                          HLT     2               ;RECEIVER NOT BLINDED, ERROR
006356  104400                  2$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP

                                        ;SET HALF DUPLEX ON LINE 2
                                        ;TRANSMITI A BINARY COUNT PATTERN
                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED

006360  012767  000340  171410  T23:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
006366  012767  000020  003272          MOV     #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
006374  012767  006470  003260          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
006402  012777  004000  003210          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
006410  004767  003312                  JSR     PC,CLRALL       ;CLEAR ALL BYTE COUNT AND
                                                                ;AND BUS ADDRESS MEMORY LOCATIONS
006414  012777  000002  003176          MOV     #2,@DHSCR       ;SELECT LINE 2
006422  012777  011754  003176          MOV     #TBUF,@DHBA     ;SET UP TO TRANSMIT
006430  012777  177400  003172          MOV     #-400,@DHBC     ;400 (OCTAL) CHARACTERS
006436  012777  073503  003160          MOV     #73503,@DHLPR   ;SET RECEIVER BLIND
                                                                ;LINE SPEED =9600 BAUD
                                                                ;CHARACTER LENGTH = 8 BITS
006444  012777  000004  003160          MOV     #4,@DHBAR       ;SET BAR BIT FOR LINE 2
006452  005777  003154          1$:     TST     @DHBAR          ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
006456  001375                          BNE     1$
006460  105777  003134                  TSTB    @DHSCR          ;WERE ANY CHARACTERS RECEIVED
006464  100001                          BPL     2$
006466  104002                          HLT     2               ;RECEIVER NOT BLINDED, ERROR
006470  104400                  2$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP

                                        ;SET HALF DUPLEX ON LINE 3
                                        ;TRANSMITI A BINARY COUNT PATTERN
                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED

006472  012767  000340  171276  T24:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
006500  012767  000020  003160          MOV     #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
006506  012767  006602  003146          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
006514  012777  004000  003076          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
006522  004767  003200                  JSR     PC,CLRALL       ;CLEAR ALL BYTE COUNT AND
                                                                ;AND BUS ADDRESS MEMORY LOCATIONS
006526  012777  000003  003064          MOV     #3,@DHSCR       ;SELECT LINE 3
006534  012777  011754  003064          MOV     #TBUF,@DHBA     ;SET UP TO TRANSMIT
006542  012777  177400  003060          MOV     #-400,@DHBC     ;400 (OCTAL) CHARACTERS
006550  012777  073503  003046          MOV     #73503,@DHLPR   ;SET RECEIVER BLIND
                                                                ;LINE SPEED =9600 BAUD
                                                                ;CHARACTER LENGTH = 8 BITS
006556  012777  000010  003046          MOV     #10,@DHBAR      ;SET BAR BIT FOR LINE 3
006564  005777  003042          1$:     TST     @DHBAR          ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
006570  001375                          BNE     1$
006572  105777  003022                  TSTB    @DHSCR          ;WERE ANY CHARACTERS RECEIVED
006576  100001                          BPL     2$
006600  104002                          HLT     2               ;RECEIVER NOT BLINDED, ERROR
006602  104400                  2$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP
```

```
                                        ;SET HALF DUPLEX ON LINE 4
                                        ;TRANSMITI A BINARY COUNT PATTERN
                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED

006604  012767  000340  171164  T25:  MOV   #340,PS              ;DISABLE ALL INTERRUPTS
006612  012767  000020  003046        MOV   #20,ICOUNT           ;SET UP FOR 20 ITERATIONS
006620  012767  006714  003034        MOV   #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
006626  012777  004000  002764        MOV   #BIT11,@DHSCR        ;MASTER CLEAR INTERFACE
006634  004767  003066                JSR   PC,CLRALL            ;CLEAR ALL BYTE COUNT AND
                                                                 ;AND BUS ADDRESS MEMORY LOCATIONS
006640  012777  000004  002752        MOV   #4,@DHSCR            ;SELECT LINE 4
006646  012777  011754  002752        MOV   #TBUF,@DHBA          ;SET UP TO TRANSMIT
006654  012777  177400  002746        MOV   #-400,@DHBC          ;400 (OCTAL) CHARACTERS
006662  012777  073503  002734        MOV   #73503,@DHLPR        ;SET RECEIVER BLIND
                                                                 ;LINE SPEED =9600 BAUD
                                                                 ;CHARACTER LENGTH = 8 BITS
006670  012777  000020  002734        MOV   #20,@DHBAR           ;SET BAR BIT FOR LINE 4
006676  005777  002730          1$:   TST   @DHBAR               ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
006702  001375                      BNE   1$
006704  105777  002710              TSTB  @DHSCR               ;WERE ANY CHARACTERS RECEIVED
006710  100001                      BPL   2$
006712  104002                      HLT   2                    ;RECEIVER NOT BLINDED, ERROR
006714  104400          2$:   SCOPE                            ;CHECK FOR ITERATIONS, LOOP

                                        ;SET HALF DUPLEX ON LINE 5
                                        ;TRANSMITI A BINARY COUNT PATTERN
                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED

006716  012767  000340  171052  T26:  MOV   #340,PS              ;DISABLE ALL INTERRUPTS
006724  012767  000020  002734        MOV   #20,ICOUNT           ;SET UP FOR 20 ITERATIONS
006732  012767  007026  002722        MOV   #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
006740  012777  004000  002652        MOV   #BIT11,@DHSCR        ;MASTER CLEAR INTERFACE
006746  004767  002754                JSR   PC,CLRALL            ;CLEAR ALL BYTE COUNT AND
                                                                 ;AND BUS ADDRESS MEMORY LOCATIONS
006752  012777  000005  002640        MOV   #5,@DHSCR            ;SELECT LINE 5
006760  012777  011754  002640        MOV   #TBUF,@DHBA          ;SET UP TO TRANSMIT
006766  012777  177400  002634        MOV   #-400,@DHBC          ;400 (OCTAL) CHARACTERS
006774  012777  073503  002622        MOV   #73503,@DHLPR        ;SET RECEIVER BLIND
                                                                 ;LINE SPEED =9600 BAUD
                                                                 ;CHARACTER LENGTH = 8 BITS
007002  012777  000040  002622        MOV   #40,@DHBAR           ;SET BAR BIT FOR LINE 5
007010  005777  002616          1$:   TST   @DHBAR               ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
007014  001375                      BNE   1$
007016  105777  002576              TSTB  @DHSCR               ;WERE ANY CHARACTERS RECEIVED
007022  100001                      BPL   2$
007024  104002                      HLT   2                    ;RECEIVER NOT BLINDED, ERROR
007026  104400          2$:   SCOPE                            ;CHECK FOR ITERATIONS, LOOP

                                        ;SET HALF DUPLEX ON LINE 6
                                        ;TRANSMITI A BINARY COUNT PATTERN
                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED

007030  012767  000340  170740  T27:  MOV   #340,PS              ;DISABLE ALL INTERRUPTS
007036  012767  000020  002622        MOV   #20,ICOUNT           ;SET UP FOR 20 ITERATIONS
007044  012767  007140  002610        MOV   #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
```

```
 1772  007052  012777  004000  002540          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1773  007060  004767  002642                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
 1774                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
 1775  007064  012777  000006  002526          MOV     #6,@DHSCR               ;SELECT LINE 6
 1776  007072  012777  011754  002526          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
 1777  007100  012777  177400  002522          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
 1778  007106  012777  073503  002510          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
 1779                                                                          ;LINE SPEED =9600 BAUD
 1780                                                                          ;CHARACTER LENGTH = 8 BITS
 1781  007114  012777  000100  002510          MOV     #100,@DHBAR             ;SET BAR BIT FOR LINE 6
 1782  007122  005777  002504          1$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
 1783  007126  001375                          BNE     1$
 1784  007130  105777  002464                  TSTB    @DHSCR                  ;WERE ANY CHARACTERS RECEIVED
 1785  007134  100001                          BPL     2$
 1786  007136  104002                          HLT     2                       ;RECEIVER NOT BLINDED, ERROR
 1787  007140  104400          2$:             SCOPE                           ;CHECK FOR ITERATIONS, LOOP
 1788
 1789                                          ;SET HALF DUPLEX ON LINE 7
 1790                                          ;TRANSMITI A BINARY COUNT PATTERN
 1791                                          ;VERIFY THAT NO CHARACTERS ARE RECEIVED
 1792
 1793  007142  012767  000340  170626  T30:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
 1794  007150  012767  000020  002510          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
 1795  007156  012767  007252  002476          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
 1796  007164  012777  004000  002426          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1797  007172  004767  002530                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
 1798                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
 1799  007176  012777  000007  002414          MOV     #7,@DHSCR               ;SELECT LINE 7
 1800  007204  012777  011754  002414          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
 1801  007212  012777  177400  002410          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
 1802  007220  012777  073503  002376          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
 1803                                                                          ;LINE SPEED =9600 BAUD
 1804                                                                          ;CHARACTER LENGTH = 8 BITS
 1805  007226  012777  000200  002376          MOV     #200,@DHBAR             ;SET BAR BIT FOR LINE 7
 1806  007234  005777  002372          1$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
 1807  007240  001375                          BNE     1$
 1808  007242  105777  002352                  TSTB    @DHSCR                  ;WERE ANY CHARACTERS RECEIVED
 1809  007246  100001                          BPL     2$
 1810  007250  104002                          HLT     2                       ;RECEIVER NOT BLINDED, ERROR
 1811  007252  104400          2$:             SCOPE                           ;CHECK FOR ITERATIONS, LOOP
 1812
 1813                                          ;SET HALF DUPLEX ON LINE 10
 1814                                          ;TRANSMITI A BINARY COUNT PATTERN
 1815                                          ;VERIFY THAT NO CHARACTERS ARE RECEIVED
 1816
 1817  007254  012767  000340  170514  T31:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
 1818  007262  012767  000020  002376          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
 1819  007270  012767  007364  002364          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
 1820  007276  012777  004000  002314          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1821  007304  004767  002416                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
 1822                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
 1823  007310  012777  000010  002302          MOV     #10,@DHSCR              ;SELECT LINE 10
 1824  007316  012777  011754  002302          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
 1825  007324  012777  177400  002276          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
 1826  007332  012777  073503  002264          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
 1827                                                                          ;LINE SPEED =9600 BAUD
```

```
                                                                ;CHARACTER LENGTH = 8 BITS
007340  012777  000400  002264          MOV     #400,@DHBAR     ;SET BAR BIT FOR LINE 10
007346  005777  002260          1$:     TST     @DHBAR          ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
007352  001375                          BNE     1$
007354  105777  002240                  TSTB    @DHSCR          ;WERE ANY CHARACTERS RECEIVED
007360  100001                          BPL     2$
007362  104002                          HLT     2               ;RECEIVER NOT BLINDED, ERROR
007364  104400                  2$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP

                                ;SET HALF DUPLEX ON LINE 11
                                ;TRANSMITI A BINARY COUNT PATTERN
                                ;VERIFY THAT NO CHARACTERS ARE RECEIVED

007366  012767  000340  170402  T32:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
007374  012767  000020  002264          MOV     #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
007402  012767  007476  002252          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
007410  012777  004000  002202          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
007416  004767  002304                  JSR     PC,CLRALL       ;CLEAR ALL BYTE COUNT AND
                                                                ;AND BUS ADDRESS MEMORY LOCATIONS
007422  012777  000011  002170          MOV     #11,@DHSCR      ;SELECT LINE 11
007430  012777  011754  002170          MOV     #TBUF,@DHBA     ;SET UP TO TRANSMIT
007436  012777  177400  002164          MOV     #-400,@DHBC     ;400 (OCTAL) CHARACTERS
007444  012777  073503  002152          MOV     #73503,@DHLPR   ;SET RECEIVER BLIND
                                                                ;LINE SPEED =9600 BAUD
                                                                ;CHARACTER LENGTH = 8 BITS
007452  012777  001000  002152          MOV     #1000,@DHBAR    ;SET BAR BIT FOR LINE 11
007460  005777  002146          1$:     TST     @DHBAR          ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
007464  001375                          BNE     1$
007466  105777  002126                  TSTB    @DHSCR          ;WERE ANY CHARACTERS RECEIVED
007472  100001                          BPL     2$
007474  104002                          HLT     2               ;RECEIVER NOT BLINDED, ERROR
007476  104400                  2$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP

                                ;SET HALF DUPLEX ON LINE 12
                                ;TRANSMITI A BINARY COUNT PATTERN
                                ;VERIFY THAT NO CHARACTERS ARE RECEIVED

007500  012767  000340  170270  T33:    MOV     #340,PS         ;DISABLE ALL INTERRUPTS
007506  012767  000020  002152          MOV     #20,ICOUNT      ;SET UP FOR 20 ITERATIONS
007514  012767  007610  002140          MOV     #2$,ESCAPE      ;SET UP TO ESCAPE TO NEXT TEST
007522  012777  004000  002070          MOV     #BIT11,@DHSCR   ;MASTER CLEAR INTERFACE
007530  004767  002172                  JSR     PC,CLRALL       ;CLEAR ALL BYTE COUNT AND
                                                                ;AND BUS ADDRESS MEMORY LOCATIONS
007534  012777  000012  002056          MOV     #12,@DHSCR      ;SELECT LINE 12
007542  012777  011754  002056          MOV     #TBUF,@DHBA     ;SET UP TO TRANSMIT
007550  012777  177400  002052          MOV     #-400,@DHBC     ;400 (OCTAL) CHARACTERS
007556  012777  073503  002040          MOV     #73503,@DHLPR   ;SET RECEIVER BLIND
                                                                ;LINE SPEED =9600 BAUD
                                                                ;CHARACTER LENGTH = 8 BITS
007564  012777  002000  002040          MOV     #2000,@DHBAR    ;SET BAR BIT FOR LINE 12
007572  005777  002034          1$:     TST     @DHBAR          ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
007576  001375                          BNE     1$
007600  105777  002014                  TSTB    @DHSCR          ;WERE ANY CHARACTERS RECEIVED
007604  100001                          BPL     2$
007606  104002                          HLT     2               ;RECEIVER NOT BLINDED, ERROR
007610  104400                  2$:     SCOPE                   ;CHECK FOR ITERATIONS, LOOP
```

```
1884
1885                                        ;SET HALF DUPLEX ON LINE 13
1886                                        ;TRANSMITI A BINARY COUNT PATTERN
1887                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED
1888
1889  007612  012767  000340  170156  T34:  MOV   #340,PS              ;DISABLE ALL INTERRUPTS
1890  007620  012767  000020  002040        MOV   #20,ICOUNT           ;SET UP FOR 20 ITERATIONS
1891  007626  012767  007722  002026        MOV   #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
1892  007634  012777  004000  001756        MOV   #BIT11,@DHSCR        ;MASTER CLEAR INTERFACE
1893  007642  004767  002060              JSR   PC,CLRALL            ;CLEAR ALL BYTE COUNT AND
1894                                                                   ;AND BUS ADDRESS MEMORY LOCATIONS
1895  007646  012777  000013  001744        MOV   #13,@DHSCR           ;SELECT LINE 13
1896  007654  012777  011754  001744        MOV   #TBUF,@DHBA          ;SET UP TO TRANSMIT
1897  007662  012777  177400  001740        MOV   #-400,@DHBC          ;400 (OCTAL) CHARACTERS
1898  007670  012777  073503  001726        MOV   #73503,@DHLPR        ;SET RECEIVER BLIND
1899                                                                   ;LINE SPEED =9600 BAUD
1900                                                                   ;CHARACTER LENGTH = 8 BITS
1901  007676  012777  004000  001726        MOV   #4000,@DHBAR         ;SET BAR BIT FOR LINE 13
1902  007704  005777  001722         1$:    TST   @DHBAR               ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
1903  007710  001375                       BNE   1$
1904  007712  105777  001702              TSTB  @DHSCR               ;WERE ANY CHARACTERS RECEIVED
1905  007716  100001                       BPL   2$
1906  007720  104002                       HLT   2                    ;RECEIVER NOT BLINDED, ERROR
1907  007722  104400         2$:    SCOPE                            ;CHECK FOR ITERATIONS, LOOP
1908
1909                                        ;SET HALF DUPLEX ON LINE 14
1910                                        ;TRANSMITI A BINARY COUNT PATTERN
1911                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED
1912
1913  007724  012767  000340  170044  T35:  MOV   #340,PS              ;DISABLE ALL INTERRUPTS
1914  007732  012767  000020  001726        MOV   #20,ICOUNT           ;SET UP FOR 20 ITERATIONS
1915  007740  012767  010034  001714        MOV   #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
1916  007746  012777  004000  001644        MOV   #BIT11,@DHSCR        ;MASTER CLEAR INTERFACE
1917  007754  004767  001746              JSR   PC,CLRALL            ;CLEAR ALL BYTE COUNT AND
1918                                                                   ;AND BUS ADDRESS MEMORY LOCATIONS
1919  007760  012777  000014  001632        MOV   #14,@DHSCR           ;SELECT LINE 14
1920  007766  012777  011754  001632        MOV   #TBUF,@DHBA          ;SET UP TO TRANSMIT
1921  007774  012777  177400  001626        MOV   #-400,@DHBC          ;400 (OCTAL) CHARACTERS
1922  010002  012777  073503  001614        MOV   #73503,@DHLPR        ;SET RECEIVER BLIND
1923                                                                   ;LINE SPEED =9600 BAUD
1924                                                                   ;CHARACTER LENGTH = 8 BITS
1925  010010  012777  010000  001614        MOV   #10000,@DHBAR        ;SET BAR BIT FOR LINE 14
1926  010016  005777  001610         1$:    TST   @DHBAR               ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
1927  010022  001375                       BNE   1$
1928  010024  105777  001570              TSTB  @DHSCR               ;WERE ANY CHARACTERS RECEIVED
1929  010030  100001                       BPL   2$
1930  010032  104002                       HLT   2                    ;RECEIVER NOT BLINDED, ERROR
1931  010034  104400         2$:    SCOPE                            ;CHECK FOR ITERATIONS, LOOP
1932
1933                                        ;SET HALF DUPLEX ON LINE 15
1934                                        ;TRANSMITI A BINARY COUNT PATTERN
1935                                        ;VERIFY THAT NO CHARACTERS ARE RECEIVED
1936
1937  010036  012767  000340  167732  T36:  MOV   #340,PS              ;DISABLE ALL INTERRUPTS
1938  010044  012767  000020  001614        MOV   #20,ICOUNT           ;SET UP FOR 20 ITERATIONS
1939  010052  012767  010146  001602        MOV   #2$,ESCAPE           ;SET UP TO ESCAPE TO NEXT TEST
```

```
 1940  010060  012777  004000  001532          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1941  010066  004767  001634                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
 1942                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
 1943  010072  012777  000015  001520          MOV     #15,@DHSCR              ;SELECT LINE 15
 1944  010100  012777  011754  001520          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
 1945  010106  012777  177400  001514          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
 1946  010114  012777  073503  001502          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
 1947                                                                          ;LINE SPEED =9600 BAUD
 1948                                                                          ;CHARACTER LENGTH = 8 BITS
 1949  010122  012777  020000  001502          MOV     #20000,@DHBAR           ;SET BAR BIT FOR LINE 15
 1950  010130  005777  001476          1$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
 1951  010134  001375                          BNE     1$
 1952  010136  105777  001456                  TSTB    @DHSCR                  ;WERE ANY CHARACTERS RECEIVED
 1953  010142  100001                          BPL     2$
 1954  010144  104002                          HLT     2                       ;RECEIVER NOT BLINDED, ERROR
 1955  010146  104400                  2$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
 1956
 1957                                          ;SET HALF DUPLEX ON LINE 16
 1958                                          ;TRANSMITI A BINARY COUNT PATTERN
 1959                                          ;VERIFY THAT NO CHARACTERS ARE RECEIVED
 1960
 1961  010150  012767  000340  167620  T37:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
 1962  010156  012767  000020  001502          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
 1963  010164  012767  010260  001470          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
 1964  010172  012777  004000  001420          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1965  010200  004767  001522                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
 1966                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
 1967  010204  012777  000016  001406          MOV     #16,@DHSCR              ;SELECT LINE 16
 1968  010212  012777  011754  001406          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
 1969  010220  012777  177400  001402          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
 1970  010226  012777  073503  001370          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
 1971                                                                          ;LINE SPEED =9600 BAUD
 1972                                                                          ;CHARACTER LENGTH = 8 BITS
 1973  010234  012777  040000  001370          MOV     #40000,@DHBAR           ;SET BAR BIT FOR LINE 16
 1974  010242  005777  001364          1$:     TST     @DHBAR                  ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
 1975  010246  001375                          BNE     1$
 1976  010250  105777  001344                  TSTB    @DHSCR                  ;WERE ANY CHARACTERS RECEIVED
 1977  010254  100001                          BPL     2$
 1978  010256  104002                          HLT     2                       ;RECEIVER NOT BLINDED, ERROR
 1979  010260  104400                  2$:     SCOPE                           ;CHECK FOR ITERATIONS, LOOP
 1980
 1981                                          ;SET HALF DUPLEX ON LINE 17
 1982                                          ;TRANSMITI A BINARY COUNT PATTERN
 1983                                          ;VERIFY THAT NO CHARACTERS ARE RECEIVED
 1984
 1985  010262  012767  000340  167506  T40:     MOV     #340,PS                 ;DISABLE ALL INTERRUPTS
 1986  010270  012767  000020  001370          MOV     #20,ICOUNT              ;SET UP FOR 20 ITERATIONS
 1987  010276  012767  010372  001356          MOV     #2$,ESCAPE              ;SET UP TO ESCAPE TO NEXT TEST
 1988  010304  012777  004000  001306          MOV     #BIT11,@DHSCR           ;MASTER CLEAR INTERFACE
 1989  010312  004767  001410                  JSR     PC,CLRALL               ;CLEAR ALL BYTE COUNT AND
 1990                                                                          ;AND BUS ADDRESS MEMORY LOCATIONS
 1991  010316  012777  000017  001274          MOV     #17,@DHSCR              ;SELECT LINE 17
 1992  010324  012777  011754  001274          MOV     #TBUF,@DHBA             ;SET UP TO TRANSMIT
 1993  010332  012777  177400  001270          MOV     #-400,@DHBC             ;400 (OCTAL) CHARACTERS
 1994  010340  012777  073503  001256          MOV     #73503,@DHLPR           ;SET RECEIVER BLIND
 1995                                                                          ;LINE SPEED =9600 BAUD
```

```
 1996
 1997  010346  012777  100000  001256          MOV     #100000,@DHBAR          ;CHARACTER LENGTH = 8 BITS
 1998  010354  005777  001252          1$:     TST     @DHBAR                  ;SET BAR BIT FOR LINE 17
 1999  010360  001375                  BNE     1$                              ;WAIT FOR ALL CHARACTERS TO BE TRANSMITTED
 2000  010362  105777  001232          TSTB    @DHSCR                          ;WERE ANY CHARACTERS RECEIVED
 2001  010366  100001                  BPL     2$
 2002  010370  104002                  HLT     2                               ;RECEIVER NOT BLINDED, ERROR
 2003  010372  104400          2$:     SCOPE                                   ;CHECK FOR ITERATIONS, LOOP
```

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

010374  104401              EOP:    TYPE                            ;TYPE NAME OF TEST
010376  013042                    MEPASS
010400  005067  001312            CLR     LAST                    ;CLEAR LAST ERROR PC
010404  005067  001242            CLR     ERRFLG                  ;CLEAR ERROR FLAG
010410  005267  001240            INC     PASCNT                  ;UPDATE PASS COUNT
010414  016767  001234  167146    MOV     PASCNT,LIGHTS           ;DISPLAY PASS COUNT
010422  013701  000042            MOV     @#42,R1                 ;CHECK FOR ACT-11 OR DDP
010426  001405                    BEQ     RESTRT                  ;IF NOT, CONTINUE TESTING
010430  000005                    RESET
010432  004711              LOGICAL:      JSR     PC,(R1)
010434  000240                    NOP
010436  000240                    NOP
010440  000240                    NOP
010442  000167  170534      RESTRT: JMP     BEGIN

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

010446  032767  002000  167114  SCOPER: BIT     #SW10,SWR
010454  001030                    BNE     4$
010456  032767  040000  167104  1$:     BIT     #SW14,SWR
010464  001021                    BNE     3$
010466  032767  004000  167074          BIT     #SW11,SWR
010474  001006                    BNE     2$
010476  005267  001166            INC     LPCNT
010502  026767  001162  001156    CMP     LPCNT,ICOUNT
010510  001007                    BNE     3$
010512  005067  001152          2$:     CLR     LPCNT
010516  005067  001130            CLR     ERRFLG
010522  011667  001132            MOV     (SP),RETURN
010526  000002                    RTI
010530  016716  001124          3$:     MOV     RETURN,(SP)
010534  000002                    RTI
010536  005767  001110          4$:     TST     ERRFLG
010542  001745                    BEQ     1$
010544  000762                    BR      2$

                                        ;CHECK FOR FREEZE ON CURRENT DATA

010546  032767  001000  167014  SCOP1R: BIT     #SW09,SWR
010554  001402                    BEQ     1$
010556  016716  001102            MOV     FREEZ1,(SP)
010562  000002                1$:     RTI
```

```
2055
2056                                            ;ERROR HANDLER
2057
2058  010564  032767  020000  166776  ERRORS:   BIT     #SW13,SWR
2059  010572  001051                            BNE     HALTS
2060  010574  021667  001116                    CMP     (SP),LAST
2061  010600  001404                            BEQ     1$
2062  010602  011667  001110                    MOV     (SP),LAST
2063  010606  005067  001040                    CLR     ERRFLG
2064  010612  104406                  1$:       SAVOSP
2065  010614  011605                            MOV     (SP),R5
2066  010616  162705  000002                    SUB     #2,R5
2067  010622  011504                            MOV     (R5),R4
2068  010624  006304                            ASL     R4
2069  010626  006304                            ASL     R4
2070  010630  042704  177001                    BIC     #177001,R4
2071  010634  062704  013152                    ADD     #ERRTAB,R4
2072  010640  012467  000034                    MOV     (R4)+,ERRMSG
2073  010644  011467  000042                    MOV     (R4),DATABP
2074  010650  005767  000776                    TST     ERRFLG
2075  010654  001403                            BEQ     TYPMSG
2076  010656  005767  000030                    TST     DATABP
2077  010662  001007                            BNE     TYPDAT
2078  010664  104402                  TYPMSG:   OCTASC
2079  010666  010760                            ERTABO
2080  010670  012767  000001  000754            MOV     #1,ERRFLG
2081  010676  104401                            TYPE
2082  010700  000000                  ERRMSG:   0
2083  010702  005767  000004          TYPDAT:   TST     DATABP
2084  010706  001402                            BEQ     RESREG
2085  010710  104402                            OCTASC
2086  010712  000000                  DATABP:   0
2087  010714  104407                  RESREG:   RESOS
2088  010716  005767  166646          HALTS:    TST     SWR
2089  010722  100005                            BPL     EXITER
2090  010724  010046                            PUSHRO
2091  010726  016600  000002                    MOV     2(SP),RO
2092  010732  000000                            HALT
2093  010734  012600                            POPRO
2094  010736  005267  000714          EXITER:   INC     ERRCNT
2095  010742  032767  002000  166620            BIT     #SW10,SWR
2096  010750  001402                            BEQ     1$
2097  010752  016716  000704                    MOV     ESCAPE,(SP)
2098  010756  000002                  1$:       RTI
2099  010760  000001                  ERTABO:   1
2100  010762     006     002                    .BYTE   6,2
2101  010764  011710                            SAVPC
```

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

010766  011646                TRPSRV: MOV     (SP),-(SP)          ;GET PC OF RETURN
010770  162716  000002                SUB     #2,(SP)             ;=PC OF TRAP
010774  017616  000000                MOV     @(SP),(SP)          ;GET TRP
011000  006316                TRPOK:  ASL     (SP)                ;MULTIPLY TRAP ARG BY 2
011002  042716  177001                BIC     #177001,(SP)        ;CLEAR UNWANTED BITS
011006  062716  013072                ADD     #TRPTAB,(SP)        ;POINTER TO SUBROUTINE ADDRESS
011012  017616  000000                MOV     @(SP),(SP)          ;SUBROUTINE ADDRESS
011016  000136                        JMP     @(SP)+              ;GO TO SUBROUTINE

                                        ;SAVE PC OF TEST THAT FAILED AND R0-R5

011020  016667  000004  000662  SV05P:  MOV     4(SP),SAVPC

                                        ;SAVE R0-R5

011026  010567  000652          SV05:   MOV     R5,SAVR5
011032  010467  000644                  MOV     R4,SAVR4
011036  010367  000636                  MOV     R3,SAVR3
011042  010267  000630                  MOV     R2,SAVR2
011046  010167  000622                  MOV     R1,SAVR1
011052  010067  000614                  MOV     R0,SAVR0
011056  000002                          RTI
                                        ;RESTORE R0-R5

011060  016700  000606          RS05:   MOV     SAVR0,R0
011064  016701  000604                  MOV     SAVR1,R1
011070  016702  000602                  MOV     SAVR2,R2
011074  016703  000600                  MOV     SAVR3,R3
011100  016704  000576                  MOV     SAVR4,R4
011104  016705  000574                  MOV     SAVR5,R5
011110  000002                          RTI
```

```
2138
2139                                            ;TELETYPE OUTPUT ROUTINE
2140
2141  011112  017605  000000          TYPER:  MOV     @(SP),R5
2142  011116  062716  000002                  ADD     #2,(SP)
2143  011122  105777  000466          1$:     TSTB    @TPCSR
2144  011126  100375                          BPL     1$
2145  011130  105715                          TSTB    (R5)
2146  011132  001001                          BNE     2$
2147  011134  000002                          RTI
2148  011136  112577  000454          2$:     MOVB    (R5)+,@TPDBR
2149  011142  000767                          BR      1$
2150
2151                                            ;ASCII STRING INPUT ROUTINE
2152
2153  011144  017667  000000  000006  INSTRG: MOV     @(SP),MSG
2154  011152  062716  000002                  ADD     #2,(SP)
2155  011156  104401                  INSTR1: TYPE
2156  011160  000000                  MSG:    0
2157  011162  012704  013114                  MOV     #INBUF,R4
2158  011166  012703  000007                  MOV     #7,R3
2159  011172  105777  000412          1$:     TSTB    @TKCSR
2160  011176  100375                          BPL     1$
2161  011200  117714  000406                  MOVB    @TKDBR,(R4)
2162  011204  142714  000200                  BICB    #200,(R4)
2163  011210  122427  000015                  CMPB    (R4)+,#15
2164  011214  001413                          BEQ     INSTR2
2165  011216  117777  000370  000372          MOVB    @TKDBR,@TPDBR
2166  011224  105777  000364          2$:     TSTB    @TPCSR
2167  011230  100375                          BPL     2$
2168  011232  005303                          DEC     R3
2169  011234  001356                          BNE     1$
2170  011236  104401                  INSTRE: TYPE
2171  011240  012746                          MQM
2172  011242  000745                          BR      INSTR1
2173  011244  000002                  INSTR2: RTI
```

```
2174
2175                                        ;CONVERT ASCII STRING TO OCTAL
2176
2177  011246  011605                PARAMS: MOV     (SP),R5
2178  011250  012567  000146               MOV     (R5)+,LOLIM
2179  011254  012567  000144               MOV     (R5)+,HILIM
2180  011260  012567  000142               MOV     (R5)+,DEVADR
2181  011264  112567  000140               MOVB    (R5)+,LOBITS
2182  011270  112567  000135               MOVB    (R5)+,ADRCNT
2183  011274  010516                       MOV     R5,(SP)
2184  011276  005005                PARAM1: CLR     R5
2185  011300  012704  013114               MOV     #INBUF,R4
2186  011304  122714  000015               CMPB    #15,(R4)
2187  011310  001420                       BEQ     PARERR
2188  011312  121427  000060         1$:   CMPB    (R4),#60
2189  011316  002415                       BLT     PARERR
2190  011320  121427  000067               CMPB    (R4),#67
2191  011324  003012                       BGT     PARERR
2192  011326  142714  000060               BICB    #60,(R4)
2193  011332  152405                       BISB    (R4)+,R5
2194  011334  122714  000015               CMPB    #15,(R4)
2195  011340  001406                       BEQ     LIMITS
2196  011342  006305                       ASL     R5
2197  011344  006305                       ASL     R5
2198  011346  006305                       ASL     R5
2199  011350  000760                       BR      1$
2200  011352  104404                PARERR: INSTER
2201  011354  000750                       BR      PARAM1
2202
2203                                        ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2204
2205  011356  020567  000042         LIMITS: CMP     R5,HILIM
2206  011362  101373                       BHI     PARERR
2207  011364  020567  000032               CMP     R5,LOLIM
2208  011370  103770                       BLO     PARERR
2209  011372  136705  000032               BITB    LOBITS,R5
2210  011376  001365                       BNE     PARERR
2211
2212                                        ;STORE NUMBER AT SPECIFIED ADDRESS
2213
2214  011400  016704  000022               MOV     DEVADR,R4
2215  011404  010524                1$:    MOV     R5,(R4)+
2216  011406  062705  000002               ADD     #2,R5
2217  011412  105367  000013               DECB    ADRCNT
2218  011416  001372                       BNE     1$
2219  011420  000002                       RTI
2220  011422  000000                LOLIM:  0
2221  011424  000000                HILIM:  0
2222  011426  000000                DEVADR: 0
2223  011430  000000                LOBITS: 0
2224          011431                ADRCNT=LOBITS+1
```

```
2225
2226                                    ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2227
2228  011432  104401              OCTASN: TYPE
2229  011434  012752                      MCRLF
2230  011436  017601  000000              MOV     @(SP),R1
2231  011442  062716  000002              ADD     #2,(SP)
2232  011446  012167  000130              MOV     (R1)+,WRDCNT
2233  011452  112167  000126      1$:     MOVB    (R1)+,CHRCNT
2234  011456  112167  000123              MOVB    (R1)+,SPACNT
2235  011462  013167  000120              MOV     @(R1)+,BINWRD
2236  011466  016704  000114      2$:     MOV     BINWRD,R4
2237  011472  116705  000106              MOVB    CHRCNT,R5
2238  011476  012700  013126              MOV     #TEMP,R0
2239  011502  010403              3$:     MOV     R4,R3
2240  011504  042703  177770              BIC     #177770,R3
2241  011510  062703  000260              ADD     #260,R3
2242  011514  110320                      MOVB    R3,(R0)+
2243  011516  006204                      ASR     R4
2244  011520  006204                      ASR     R4
2245  011522  006204                      ASR     R4
2246  011524  005305                      DEC     R5
2247  011526  001365                      BNE     3$
2248  011530  012703  013140              MOV     #MDATA,R3
2249  011534  114023              4$:     MOVB    -(R0),(R3)+
2250  011536  105367  000042              DECB    CHRCNT
2251  011542  001374                      BNE     4$
2252  011544  105767  000035              TSTB    SPACNT
2253  011550  001405                      BEQ     6$
2254  011552  112723  000240      5$:     MOVB    #240,(R3)+
2255  011556  105367  000023              DECB    SPACNT
2256  011562  001373                      BNE     5$
2257  011564  105013              6$:     CLRB    (R3)
2258  011566  104401                      TYPE
2259  011570  013140                      MDATA
2260  011572  005367  000004              DEC     WRDCNT
2261  011576  001325                      BNE     1$
2262  011600  000002                      RTI
2263  011602  000000              WRDCNT: 0
2264  011604  000000              CHRCNT: 0
2265          011605              SPACNT=CHRCNT+1
2266  011606  000000              BINWRD: 0
```

```
2267                                    ;INDIRECT POINTERS
2268
2269  011610  177560            TKCSR:  177560
2270  011612  177562            TKDBR:  177562
2271  011614  177564            TPCSR:  177564
2272  011616  177566            TPDBR:  177566
2273  011620  000000            DHSCR:  0
2274  011622  000000            DHNRC:  0
2275  011624  000000            DHLPR:  0
2276  011626  000000            DHBA:   0
2277  011630  000000            DHBC:   0
2278  011632  000000            DHBAR:  0
2279  011634  000000            DHBCR:  0
2280  011636  000000            DHSSR:  0
2281  011640  000000            DHSLR:  0
2282  011642  000000            DHRVEC: 0
2283  011644  000000            DHRLVL: 0
2284  011646  000000            DHTVEC: 0
2285  011650  000000            DHTLVL: 0
2286                                    ;PROGRAM VARIABLES
2287
2288  011652  000000            ERRFLG: 0               ;ERROR FLAG
2289  011654  000000            PASCNT: 0               ;PASS COUNT
2290  011656  000000            ERRCNT: 0               ;ERROR COUNT
2291  011660  000000            RETURN: 0               ;SCOPE RETURN ADDRESS FOR TEST LOOPING
2292  011662  000000            ESCAPE: 0               ;ADDRESS FOR ERROR ESCAPE
2293  011664  000000            FREEZI: 0               ;DATA LOOPING RETURN ADDRESS
2294  011666  000000            ICOUNT: 0               ;ITERATION COUNT FOR TEST IN PROGRESS
2295  011670  000000            LPCNT:  0               ;NUMBER OF ITERATIONS THIS TEST
2296  011672  000000            SAVR0:  0               ;R0 SAVE AREA
2297  011674  000000            SAVR1:  0               ;R1 SAVE AREA
2298  011676  000000            SAVR2:  0               ;R2 SAVE AREA
2299  011700  000000            SAVR3:  0               ;R3 SAVE ARE
2300  011702  000000            SAVR4:  0               ;R4 SAVE AREA
2301  011704  000000            SAVR5:  0               ;R5 SAVE AREA
2302  011706  000000            SAVSP:  0               ;STACK POINTER SAVE AREA
2303  011710  000000            SAVPC:  0               ;CALLING ROUTINE SAVE AREA
2304  011712  000000            INIFLG: 0               ;PROGRAM INITIALIZATION FLAG
2305  011714  000000            STFLG:  0               ;PROGRAM START FLAG
2306  011716  000000            LAST:   0               ;LAST ERROR PC
2307  011720  000000            ENDFLG: 0
2308  011722  000000            NULL:   0
2309  011724  000000            LINACT: 0
```

```
2310
2311                                            ;CLEAR ALL BYTE COUNT AND BUS ADDRESS REGISTERS
2312
2313  011726  012700  000020          CLRALL: MOV     #20,R0
2314  011732  005077  177670          1$:     CLR     @DHBW
2315  011736  005077  177666                  CLR     @DHBC
2316  011742  005277  177652                  INC     @DHSCR
2317  011746  005300                          DEC     R0
2318  011750  001370                          BNE     1$
2319  011752  000207                          RTS     PC
2320  011754     000          TBUF:   .BYTE   0
2321  011755     001                  .BYTE   TDAT
2322  011756     002                  .BYTE   TDAT
2323  011757     003                  .BYTE   TDAT
2324  011760     004                  .BYTE   TDAT
2325  011761     005                  .BYTE   TDAT
2326  011762     006                  .BYTE   TDAT
2327  011763     007                  .BYTE   TDAT
2328  011764     010                  .BYTE   TDAT
2329  011765     011                  .BYTE   TDAT
2330  011766     012                  .BYTE   TDAT
2331  011767     013                  .BYTE   TDAT
2332  011770     014                  .BYTE   TDAT
2333  011771     015                  .BYTE   TDAT
2334  011772     016                  .BYTE   TDAT
2335  011773     017                  .BYTE   TDAT
2336  011774     020                  .BYTE   TDAT
2337  011775     021                  .BYTE   TDAT
2338  011776     022                  .BYTE   TDAT
2339  011777     023                  .BYTE   TDAT
2340  012000     024                  .BYTE   TDAT
2341  012001     025                  .BYTE   TDAT
2342  012002     026                  .BYTE   TDAT
2343  012003     027                  .BYTE   TDAT
2344  012004     030                  .BYTE   TDAT
2345  012005     031                  .BYTE   TDAT
2346  012006     032                  .BYTE   TDAT
2347  012007     033                  .BYTE   TDAT
2348  012010     034                  .BYTE   TDAT
2349  012011     035                  .BYTE   TDAT
2350  012012     036                  .BYTE   TDAT
2351  012013     037                  .BYTE   TDAT
2352  012014     040                  .BYTE   TDAT
2353  012015     041                  .BYTE   TDAT
2354  012016     042                  .BYTE   TDAT
2355  012017     043                  .BYTE   TDAT
2356  012020     044                  .BYTE   TDAT
2357  012021     045                  .BYTE   TDAT
2358  012022     046                  .BYTE   TDAT
2359  012023     047                  .BYTE   TDAT
2360  012024     050                  .BYTE   TDAT
2361  012025     051                  .BYTE   TDAT
2362  012026     052                  .BYTE   TDAT
2363  012027     053                  .BYTE   TDAT
2364  012030     054                  .BYTE   TDAT
2365  012031     055                  .BYTE   TDAT
```

```
 2366  012032      056                    .BYTE   TDAT
 2367  012033      057                    .BYTE   TDAT
 2368  012034      060                    .BYTE   TDAT
 2369  012035      061                    .BYTE   TDAT
 2370  012036      062                    .BYTE   TDAT
 2371  012037      063                    .BYTE   TDAT
 2372  012040      064                    .BYTE   TDAT
 2373  012041      065                    .BYTE   TDAT
 2374  012042      066                    .BYTE   TDAT
 2375  012043      067                    .BYTE   TDAT
 2376  012044      070                    .BYTE   TDAT
 2377  012045      071                    .BYTE   TDAT
 2378  012046      072                    .BYTE   TDAT
 2379  012047      073                    .BYTE   TDAT
 2380  012050      074                    .BYTE   TDAT
 2381  012051      075                    .BYTE   TDAT
 2382  012052      076                    .BYTE   TDAT
 2383  012053      077                    .BYTE   TDAT
 2384  012054      100                    .BYTE   TDAT
 2385  012055      101                    .BYTE   TDAT
 2386  012056      102                    .BYTE   TDAT
 2387  012057      103                    .BYTE   TDAT
 2388  012060      104                    .BYTE   TDAT
 2389  012061      105                    .BYTE   TDAT
 2390  012062      106                    .BYTE   TDAT
 2391  012063      107                    .BYTE   TDAT
 2392  012064      110                    .BYTE   TDAT
 2393  012065      111                    .BYTE   TDAT
 2394  012066      112                    .BYTE   TDAT
 2395  012067      113                    .BYTE   TDAT
 2396  012070      114                    .BYTE   TDAT
 2397  012071      115                    .BYTE   TDAT
 2398  012072      116                    .BYTE   TDAT
 2399  012073      117                    .BYTE   TDAT
 2400  012074      120                    .BYTE   TDAT
 2401  012075      121                    .BYTE   TDAT
 2402  012076      122                    .BYTE   TDAT
 2403  012077      123                    .BYTE   TDAT
 2404  012100      124                    .BYTE   TDAT
 2405  012101      125                    .BYTE   TDAT
 2406  012102      126                    .BYTE   TDAT
 2407  012103      127                    .BYTE   TDAT
 2408  012104      130                    .BYTE   TDAT
 2409  012105      131                    .BYTE   TDAT
 2410  012106      132                    .BYTE   TDAT
 2411  012107      133                    .BYTE   TDAT
 2412  012110      134                    .BYTE   TDAT
 2413  012111      135                    .BYTE   TDAT
 2414  012112      136                    .BYTE   TDAT
 2415  012113      137                    .BYTE   TDAT
 2416  012114      140                    .BYTE   TDAT
 2417  012115      141                    .BYTE   TDAT
 2418  012116      142                    .BYTE   TDAT
 2419  012117      143                    .BYTE   TDAT
 2420  012120      144                    .BYTE   TDAT
 2421  012121      145                    .BYTE   TDAT
```

```
2422  012122     146                    .BYTE   TDAT
2423  012123     147                    .BYTE   TDAT
2424  012124     150                    .BYTE   TDAT
2425  012125     151                    .BYTE   TDAT
2426  012126     152                    .BYTE   TDAT
2427  012127     153                    .BYTE   TDAT
2428  012130     154                    .BYTE   TDAT
2429  012131     155                    .BYTE   TDAT
2430  012132     156                    .BYTE   TDAT
2431  012133     157                    .BYTE   TDAT
2432  012134     160                    .BYTE   TDAT
2433  012135     161                    .BYTE   TDAT
2434  012136     162                    .BYTE   TDAT
2435  012137     163                    .BYTE   TDAT
2436  012140     164                    .BYTE   TDAT
2437  012141     165                    .BYTE   TDAT
2438  012142     166                    .BYTE   TDAT
2439  012143     167                    .BYTE   TDAT
2440  012144     170                    .BYTE   TDAT
2441  012145     171                    .BYTE   TDAT
2442  012146     172                    .BYTE   TDAT
2443  012147     173                    .BYTE   TDAT
2444  012150     174                    .BYTE   TDAT
2445  012151     175                    .BYTE   TDAT
2446  012152     176                    .BYTE   TDAT
2447  012153     177                    .BYTE   TDAT
2448  012154     200                    .BYTE   TDAT
2449  012155     201                    .BYTE   TDAT
2450  012156     202                    .BYTE   TDAT
2451  012157     203                    .BYTE   TDAT
2452  012160     204                    .BYTE   TDAT
2453  012161     205                    .BYTE   TDAT
2454  012162     206                    .BYTE   TDAT
2455  012163     207                    .BYTE   TDAT
2456  012164     210                    .BYTE   TDAT
2457  012165     211                    .BYTE   TDAT
2458  012166     212                    .BYTE   TDAT
2459  012167     213                    .BYTE   TDAT
2460  012170     214                    .BYTE   TDAT
2461  012171     215                    .BYTE   TDAT
2462  012172     216                    .BYTE   TDAT
2463  012173     217                    .BYTE   TDAT
2464  012174     220                    .BYTE   TDAT
2465  012175     221                    .BYTE   TDAT
2466  012176     222                    .BYTE   TDAT
2467  012177     223                    .BYTE   TDAT
2468  012200     224                    .BYTE   TDAT
2469  012201     225                    .BYTE   TDAT
2470  012202     226                    .BYTE   TDAT
2471  012203     227                    .BYTE   TDAT
2472  012204     230                    .BYTE   TDAT
2473  012205     231                    .BYTE   TDAT
2474  012206     232                    .BYTE   TDAT
2475  012207     233                    .BYTE   TDAT
2476  012210     234                    .BYTE   TDAT
2477  012211     235                    .BYTE   TDAT
```

```
 2478  012212      236                    .BYTE   TDAT
 2479  012213      237                    .BYTE   TDAT
 2480  012214      240                    .BYTE   TDAT
 2481  012215      241                    .BYTE   TDAT
 2482  012216      242                    .BYTE   TDAT
 2483  012217      243                    .BYTE   TDAT
 2484  012220      244                    .BYTE   TDAT
 2485  012221      245                    .BYTE   TDAT
 2486  012222      246                    .BYTE   TDAT
 2487  012223      247                    .BYTE   TDAT
 2488  012224      250                    .BYTE   TDAT
 2489  012225      251                    .BYTE   TDAT
 2490  012226      252                    .BYTE   TDAT
 2491  012227      253                    .BYTE   TDAT
 2492  012230      254                    .BYTE   TDAT
 2493  012231      255                    .BYTE   TDAT
 2494  012232      256                    .BYTE   TDAT
 2495  012233      257                    .BYTE   TDAT
 2496  012234      260                    .BYTE   TDAT
 2497  012235      261                    .BYTE   TDAT
 2498  012236      262                    .BYTE   TDAT
 2499  012237      263                    .BYTE   TDAT
 2500  012240      264                    .BYTE   TDAT
 2501  012241      265                    .BYTE   TDAT
 2502  012242      266                    .BYTE   TDAT
 2503  012243      267                    .BYTE   TDAT
 2504  012244      270                    .BYTE   TDAT
 2505  012245      271                    .BYTE   TDAT
 2506  012246      272                    .BYTE   TDAT
 2507  012247      273                    .BYTE   TDAT
 2508  012250      274                    .BYTE   TDAT
 2509  012251      275                    .BYTE   TDAT
 2510  012252      276                    .BYTE   TDAT
 2511  012253      277                    .BYTE   TDAT
 2512  012254      300                    .BYTE   TDAT
 2513  012255      301                    .BYTE   TDAT
 2514  012256      302                    .BYTE   TDAT
 2515  012257      303                    .BYTE   TDAT
 2516  012260      304                    .BYTE   TDAT
 2517  012261      305                    .BYTE   TDAT
 2518  012262      306                    .BYTE   TDAT
 2519  012263      307                    .BYTE   TDAT
 2520  012264      310                    .BYTE   TDAT
 2521  012265      311                    .BYTE   TDAT
 2522  012266      312                    .BYTE   TDAT
 2523  012267      313                    .BYTE   TDAT
 2524  012270      314                    .BYTE   TDAT
 2525  012271      315                    .BYTE   TDAT
 2526  012272      316                    .BYTE   TDAT
 2527  012273      317                    .BYTE   TDAT
 2528  012274      320                    .BYTE   TDAT
 2529  012275      321                    .BYTE   TDAT
 2530  012276      322                    .BYTE   TDAT
 2531  012277      323                    .BYTE   TDAT
 2532  012300      324                    .BYTE   TDAT
 2533  012301      325                    .BYTE   TDAT
```

```
  2534  012302      326                   .BYTE   TDAT
  2535  012303      327                   .BYTE   TDAT
  2536  012304      330                   .BYTE   TDAT
  2537  012305      331                   .BYTE   TDAT
  2538  012306      332                   .BYTE   TDAT
  2539  012307      333                   .BYTE   TDAT
  2540  012310      334                   .BYTE   TDAT
  2541  012311      335                   .BYTE   TDAT
  2542  012312      336                   .BYTE   TDAT
  2543  012313      337                   .BYTE   TDAT
  2544  012314      340                   .BYTE   TDAT
  2545  012315      341                   .BYTE   TDAT
  2546  012316      342                   .BYTE   TDAT
  2547  012317      343                   .BYTE   TDAT
  2548  012320      344                   .BYTE   TDAT
  2549  012321      345                   .BYTE   TDAT
  2550  012322      346                   .BYTE   TDAT
  2551  012323      347                   .BYTE   TDAT
  2552  012324      350                   .BYTE   TDAT
  2553  012325      351                   .BYTE   TDAT
  2554  012326      352                   .BYTE   TDAT
  2555  012327      353                   .BYTE   TDAT
  2556  012330      354                   .BYTE   TDAT
  2557  012331      355                   .BYTE   TDAT
  2558  012332      356                   .BYTE   TDAT
  2559  012333      357                   .BYTE   TDAT
  2560  012334      360                   .BYTE   TDAT
  2561  012335      361                   .BYTE   TDAT
  2562  012336      362                   .BYTE   TDAT
  2563  012337      363                   .BYTE   TDAT
  2564  012340      364                   .BYTE   TDAT
  2565  012341      365                   .BYTE   TDAT
  2566  012342      366                   .BYTE   TDAT
  2567  012343      367                   .BYTE   TDAT
  2568  012344      370                   .BYTE   TDAT
  2569  012345      371                   .BYTE   TDAT
  2570  012346      372                   .BYTE   TDAT
  2571  012347      373                   .BYTE   TDAT
  2572  012350      374                   .BYTE   TDAT
  2573  012351      375                   .BYTE   TDAT
  2574  012352      376                   .BYTE   TDAT
  2575  012353      377                   .BYTE   TDAT
  2576                                    .EVEN
  2577  012354  000000                    RBUF:   0
  2578          012416                    .=.+40
```

```
  2579  012416  120000                RWRD0:  120000
  2580  012420  120400                RWRD1:  120400
  2581  012422  121000                RWRD2:  121000
  2582  012424  121400                RWRD3:  121400
  2583  012426  122000                RWRD4:  122000
  2584  012430  122400                RWRD5:  122400
  2585  012432  123000                RWRD6:  123000
  2586  012434  123400                RWRD7:  123400
  2587  012436  124000                RWRD10: 124000
  2588  012440  124400                RWRD11: 124400
  2589  012442  125000                RWRD12: 125000
  2590  012444  125400                RWRD13: 125400
  2591  012446  126000                RWRD14: 126000
  2592  012450  126400                RWRD15: 126400
  2593  012452  127000                RWRD16: 127000
  2594  012454  127400                RWRD17: 127400
```

```
2595                                            ;ENTER HERE ON POWER FAILURE
2596
2597
2598  012456  010046                  PFAIL:  MOV     R0,-(SP)                ;SAVE R0-R5 ON PROCESSOR STACK
2599  012460  010146                          MOV     R1,-(SP)
2600  012462  010246                          MOV     R2,-(SP)
2601  012464  010346                          MOV     R3,-(SP)
2602  012466  010446                          MOV     R4,-(SP)
2603  012470  010546                          MOV     R5,-(SP)
2604  012472  016746  165326                  MOV     24,-(SP)
2605  012476  010667  177204                  MOV     SP,SAVSP                ;SAVE STACK POINTER
2606  012502  012767  012514  165314          MOV     #RESTART,24             ;SET UP FOR POWER UP TRAP
2607  012510  000000                          HALT                            ;HALT ON POWER DOWN NORMAL
2608  012512  000777                          BR      .
2609
2610                                            ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2611
2612  012514  016706  177166          RESTAR: MOV     SAVSP,SP                ;RESTORE STACK POINTER
2613  012520  012605                          MOV     (SP)+,R5                ;RESTORE R0-R5
2614  012522  012604                          MOV     (SP)+,R4
2615  012524  012603                          MOV     (SP)+,R3
2616  012526  012602                          MOV     (SP)+,R2
2617  012530  012601                          MOV     (SP)+,R1
2618  012532  012600                          MOV     (SP)+,R0
2619  012534  012767  012456  165262          MOV     #PFAIL,24               ;SET UP FOR POWER FAILURE
2620  012542  012767  000340  165226          MOV     #340,PS
2621  012550  012706  013540                  MOV     #STACK,SP
2622  012554  005067  000346                  CLR     TEMP
2623  012560  005267  000342                  INC     TEMP
2624  012564  001375                          BNE     .-4
2625  012566  104402                          OCTASC
2626  012570  012612                          PFTAB
2627  012572  104401                          TYPE
2628  012574  012755                          MPFAIL
2629  012576  005067  177050                  CLR     ERRFLG
2630  012602  005067  177110                  CLR     LAST
2631  012606  000177  177046                  JMP     @RETURN
2632  012612  000001                  PFTAB:  1
2633  012614  000006  000002                  6,2
2634  012620  000207                          RETURN
```

```
  2635  012622  005015  042012  030510  MTITLE: .ASCIZ  <15><12><12>/DH11 HALF-DUPLEX AND BREAK TEST /<15><12>
  2636  012630  020061  040510  043114
  2637  012636  042055  050125  042514
  2638  012644  020130  047101  020104
  2639  012652  051102  040505  020113
  2640  012660  042524  052123  006440
  2641  012666  000012
  2642  012670  005015  042526  052103  MVECTO: .ASCIZ  <15><12>/VECTOR ADDRESS-/
  2643  012676  051117  040440  042104
  2644  012704  042522  051523  000055
  2645  012712  005015  047503  052116  MREGAD: .ASCIZ  <15><12>/CONTROL REGISTER ADDRESS-/
  2646  012720  047522  020114  042522
  2647  012726  044507  052123  051105
  2648  012734  040440  042104  042522
  2649  012742  051523  000055
  2650  012746  020040  000077          MQM:    .ASCIZ  /  ?/
  2651  012752  005015     000          MCRLF:  .ASCIZ   <15><12>
  2652  012755     040  050040  053517  MPFAIL: .ASCIZ  /  POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
  2653  012762  051105  043040  044501
  2654  012770  052514  042522  020054
  2655  012776  051120  043517  040522
  2656  013004  020115  042522  052123
  2657  013012  051101  020124  052101
  2658  013020  052040  051505  020124
  2659  013026  047111  050040  047522
  2660  013034  051107  051505  000123
  2661  013042  005015  055104  044104  MEPASS: .ASCIZ  <15><12>/DZDHI/
  2662  013050  000111
  2663  013052  005015  000122          MR:     .ASCIZ  <15><12>/R/
  2664  013056  005015  042524  052123  MTSTPC: .ASCIZ  <15><12>/TEST PC-/
  2665  013064  050040  026503     000
  2666          013072                  .EVEN
  2667
  2668                                  ;TABLE OF POINTERS FOR TRAP DECODING
  2669
  2670  013072  010446                  TRPTAB: SCOPER
  2671  013074  011112                          TYPER
  2672  013076  011432                          OCTASN
  2673  013100  011144                          INSTRG
  2674  013102  011236                          INSTRE
  2675  013104  011246                          PARAMS
  2676  013106  011020                          SVOSP
  2677  013110  011060                          RSOS
  2678  013112  010546                          SCOP1R
  2679
  2680                                  ;BUFFERS FOR INPUT-OUTPUT
  2681
  2682  013114  000000                  INBUF:  0
  2683          013126                  .=.+10
  2684  013126  000000                  TEMP:   0
  2685          013140                  .=.+10
  2686  013140  000000                  MDATA:  0
  2687          013152                  .=.+10
  2688
  2689                                  ;TABLE OF POINTERS TO ERROR MESSAGES AND DATA
  2690
```

```
2691  013152                          ERRTAB:
2692  013152  013166                          EM1
2693  013154  013326                          DT1
2694  013156  013242                          EM2
2695  013160  013326                          DT1
2696  013162  013300                          EM3
2697  013164  000000                          0
2698  013166  047515  042522  052040  EM1:    .ASCIZ  /MORE THAN 1 CHARACTER RECEIVED/<15><12>/EXP     REC/
2699  013174  040510  020116  020061
2700  013202  044103  051101  041501
2701  013210  042524  020122  042522
2702  013216  042503  053111  042105
2703  013224  005015  054105  020120
2704  013232  020040  020040  042522
2705  013240  000103
2706  013242  051102  040505  020113  EM2:    .ASCIZ  /BREAK DATA ERROR/<15><12>/EXP     REC/
2707  013250  040504  040524  042440
2708  013256  051122  051117  005015
2709  013264  054105  020120  020040
2710  013272  020040  042522  000103
2711  013300  042522  042503  053111  EM3:    .ASCIZ  /RECEIVER NOT BLINDED/
2712  013306  051105  047040  052117
2713  013314  041040  044514  042116
2714  013322  042105     000
2715          013326                  .EVEN
2716  013326  000002                  DT1:    2
2717  013330     006     002          .BYTE   6,2
2718  013332  011704                          SAVR5
2719  013334     006     002          .BYTE   6,2
2720  013336  011702                          SAVR4
2721  013340  000000                  ENDCOD: 0
2722          000001                  .END
```

```
ADRCNT= 011431           2182*  2217*  2224#
BEGIN   001202            877    906    912#  2025
BINWRD  011606           2235*  2236   2266#
BITX  = 000000            932#   975#  1018#  1061#  1104#  1147#  1190#  1233#  1276#  1319#  1362#  1405#  1448#
                         1491#  1534#  1577#  1620#  1644#  166A#  1692#  1716#  1740#  1764#  1788#  1812#  1836#
                         1860#  1884#  1908#  1932#  1956#  1980#  2004#
BIT00 = 000001            563#
BIT01 = 000002            562#
BIT02 = 000004            561#
BIT03 = 000010            560#
BIT04 = 000020            559#
BIT05 = 000040            558#
BIT06 = 000100            557#
BIT07 = 000200            556#
BIT08 = 000400            555#
BIT09 = 001000            554#
BIT10 = 002000            553#
BIT11 = 004000            552#   943    954    986    997   1029   1040   1072   1083   1115   1126   1158   1169
                         1201   1212   1244   1255   1287   1298   1330   1341   1373   1384   1416   1427   1459
                         1470   1502   1513   1545   1556   1588   1599   1628   1652   1676   1700   1724   1748
                         1772   1796   1820   1844   1868   1892   1916   1940   1964   1988
BIT12 = 010000            551#
BIT13 = 020000            550#
BIT14 = 040000            549#
BIT15 = 100000            548#
CHRCNT  011604           2233*  2237   2250*  2264#  2265
CLRALL  011726            944    987   1030   1073   1116   1159   1202   1245   1288   1331   1374   1417   1460
                         1503   1546   1589   1629   1653   1677   1701   1725   1749   1773   1797   1821   1845
                         1869   1893   1917   1941   1965   1989   2313#
DATA  = 127400           2579#
DATABP  010712           2073*  2076   2083   2086#
DATAX = 127400           2579#  2580#  2581#  2582#  2583#  2584#  2585#  2586#  2587#  2588#  2589#  2590#  2591#
                         2592#  2593#  2594#
DEVADR  011426           2180*  2214   2222#
DHBA    011626            947*   956*   990*   999*  1033*  1042*  1076*  1085*  1119*  1128*  1162*  1171*  1205*
                         1214*  1248*  1257*  1291*  1300*  1334*  1343*  1377*  1386*  1420*  1429*  1463*  1472*
                         1506*  1515*  1549*  1558*  1592*  1601*  1632*  1656*  1680*  1704*  1728*  1752*  1776*
                         1800*  1824*  1848*  1872*  1896*  1920*  1944*  1968*  1992*  2276#  2314*
DHBAR   011632            951*   960*   961    994*  1003*  1004   1037*  1046*  1047   1080*  1089*  1090   1123*
                         1132*  1133   1166*  1175*  1176   1209*  1218*  1219   1252*  1261*  1262   1295*  1304*
                         1305   1338*  1347*  1348   1381*  1390*  1391   1424*  1433*  1434   1467*  1476*  1477
                         1510*  1519*  1520   1553*  1562*  1563   1596*  1605*  1606   1637*  1638   1661*  1662
                         1685*  1686   1709*  1710   1733*  1734   1757*  1758   1781*  1782   1805*  1806   1829*
                         1830   1853*  1854   1877*  1878   1901*  1902   1925*  1926   1949*  1950   1973*  1974
                         1997*  1998   2278#
DHBC    011630            948*   957*   991*  1000*  1034*  1043*  1077*  1086*  1120*  1129*  1163*  1172*  1206*
                         1215*  1249*  1258*  1292*  1301*  1335*  1344*  1378*  1387*  1421*  1430*  1464*  1473*
                         1507*  1516*  1550*  1559*  1593*  1602*  1633*  1657*  1681*  1705*  1729*  1753*  1777*
                         1801*  1825*  1849*  1873*  1897*  1921*  1945*  1969*  1993*  2277#  2315*
DHBCR   011634            959*  1002*  1045*  1088*  1131*  1174*  1217*  1260*  1303*  1346*  1389*  1432*  1475*
                         1518*  1561*  1604*  2279#
DHLPR   011624            949*   958*   992*  1001*  1035*  1044*  1078*  1087*  1121*  1130*  1164*  1173*  1207*
                         1216*  1250*  1259*  1293*  1302*  1336*  1345*  1379*  1388*  1422*  1431*  1465*  1474*
                         1508*  1517*  1551*  1560*  1594*  1603*  1634*  1658*  1682*  1706*  1730*  1754*  1778*
                         1802*  1826*  1850*  1874*  1898*  1922*  1946*  1970*  1994*  2275#
DHNRC   011622            969   1012   1055   1098   1141   1184   1227   1270   1313   1356   1399   1442   1485
```

```
                    1528    1571    1614    2274#
DHRLVL  011644      2283#
DHRVEC  011642       892    2282#
DHSCR   011620       900     943*    946*    954*    955*    986*    989*    997*    998*   1029*   1032*   1040*   1041*
                    1072*   1075*   1083*   1084*   1115*   1118*   1126*   1127*   1158*   1161*   1169*   1170*   1201*
                    1204*   1212*   1213*   1244*   1247*   1255*   1256*   1287*   1290*   1298*   1299*   1330*   1333*
                    1341*   1342*   1373*   1376*   1384*   1385*   1416*   1419*   1427*   1428*   1459*   1462*   1470*
                    1471*   1502*   1505*   1513*   1514*   1545*   1548*   1556*   1557*   1588*   1591*   1599*   1600*
                    1628*   1631*   1640    1652*   1655*   1664    1676*   1679*   1688    1700*   1703*   1712    1724*
                    1727*   1736    1748*   1751*   1760    1772*   1775*   1784    1796*   1799*   1808    1820*   1823*
                    1832    1844*   1847*   1856    1868*   1871*   1880    1892*   1895*   1904    1916*   1919*   1928
                    1940*   1943*   1952    1964*   1967*   1976    1988*   1991*   2000    2273#   2316*
DHSLR   011640       903*    904*    952     963     995    1006    1038    1049    1081    1092    1124    1135    1167
                    1178    1210    1221    1253    1264    1296    1307    1339    1350    1382    1393    1425    1436
                    1468    1479    1511    1522    1554    1565    1597    1608    2281#
DHSSR   011636       903     965    1008    1051    1094    1137    1180    1223    1266    1309    1352    1395    1438
                    1481    1524    1567    1610    2280#
DHTLVL  011650      2285#
DHTVEC  011646      2284#
DT1     013326      2693    2695    2716#
EM1     013166      2692    2698#
EM2     013242      2694    2706#
EM3     013300      2696    2711#
ENDCOD  013340       535    2721#
ENDFLG  011720      2307#
EOP     010374      2012#
ERRCNT  011656       868*   2094*   2290#
ERRFLG  011652       869*    870*   2015*   2040*   2045    2063*   2074    2080*   2288#   2629*
ERRMSG  010700      2072*   2082#
ERRORS  010564       828    2058#
ERRTAB  013152      2071    2691#
ERTABO  010760      2079    2099#
ESCAPE  011662       942*    985*   1028*   1071*   1114*   1157*   1200*   1243*   1286*   1329*   1372*   1415*   1458*
                    1501*   1544*   1587*   1627*   1651*   1675*   1699*   1723*   1747*   1771*   1795*   1819*   1843*
                    1867*   1891*   1915*   1939*   1963*   1987*   2097    2292#
EXITER  010736      2089    2094#
FREEZI  011664      2053    2293#
HALTS   010716      2059    2088#
HILIM   011424      2179*   2205    2221#
ICOUNT  011666       941*    984*   1027*   1070*   1113*   1156*   1199*   1242*   1285*   1328*   1371*   1414*   1457*
                    1500*   1543*   1586*   1626*   1650*   1674*   1698*   1722*   1746*   1770*   1794*   1818*   1842*
                    1866*   1890*   1914*   1938*   1962*   1986*   2037    2294#
INBUF   013114      2157    2185    2682#
INIFLG  011712       872     905     907*   2304#
INSTER= 104404       845#   2200
INSTR = 104403       844#    887     895     916
INSTRE  011236      2170#   2674
INSTRG  011144      2153#   2673
INSTR1  011156      2155#   2172
INSTR2  011244      2164    2173#
K       = 000020     932#   1620#   2579#
KX      = 000020     932#    975#   1018#   1061#   1104#   1147#   1190#   1233#   1276#   1319#   1362#   1405#   1448#
                    1491#   1534#   1577#   1620#   1644#   1668#   1692#   1716#   1740#   1764#   1788#   1812#   1836#
                    1860#   1884#   1908#   1932#   1956#   1980#   2004#   2579#   2580#   2581#   2582#   2583#   2584#
                    2585#   2586#   2587#   2588#   2589#   2590#   2591#   2592#   2593#   2594#   2595#
LAST    011716      2014*   2060    2062*   2306#   2630*
```

```
LIGHTS= 177570           533#   2017*
LIMITS  011356          2195    2205#
LINACT  011724          2309#
LINE   = 000020           932#    975#   1018#   1061#   1104#   1147#   1190#   1233#   1276#   1319#   1362#   1405#   1448#
                         1491#   1534#   1577#   1620#   1644#   1668#   1692#   1716#   1740#   1764#   1788#   1812#   1836#
                         1860#   1884#   1908#   1932#   1956#   1980#   2004#
LOBITS  011430          2181*   2209    2223#   2224
LOGICA  010432           851    2021#
LOLIM   011422          2178*   2207    2220#
LPCNT   011670          2036*   2037    2039*   2295#
MCRLF   012752          2229    2651#
MDATA   013140          2248    2259    2686#
MEPASS  013042          2013    2661#
MPFAIL  012755          2628    2652#
MQM     012746          2171    2650#
MR      013052           929    2663#
MREGAD  012712           896    2645#
MSG     011160          2153*   2156#
MTITLE  012622           871    2635#
MTSTPC  013056           917    2664#
MVECTO  012670           888    2642#
N      = 000001             1#
NULL    011722           947     990    1033    1076    1119    1162    1205    1248    1291    1334    1377    1420    1463
                        1506    1549    1592    2308#
OCTASC= 104402           843#   2078    2085    2625
OCTASN  011432          2228#   2672
PARAM  = 104405           846#    889     897     918
PARAMS  011246          2177#   2675
PARAM1  011276          2184#   2201
PARERR  011352          2187    2189    2191    2200#   2206    2208    2210
PASCNT  011654           867*   2016*   2017    2289#
PFAIL   012456           826     865    2598#   2619
PFTAB   012612          2626    2632#
POPRO  = 012600           542#   2093
POP1SP= 005726            540#
POP2SP= 022626            544#
PS     = 177776           534#    863*    912*    940*    983*   1026*   1069*   1112*   1155*   1198*   1241*   1284*   1327*
                        1370*   1413*   1456*   1499*   1542*   1585*   1625*   1649*   1673*   1697*   1721*   1745*   1769*
                        1793*   1817*   1841*   1865*   1889*   1913*   1937*   1961*   1985*   2620*
PUSHRO= 010046           541#   2090
PUSH1S= 005746           539#
PUSH2S= 024646           543#
RBUF    012354          2577#
RESREG  010714          2084    2087#
RESTAR  012514          2606    2612#
RESTRT  010442          2019    2025#
RESOS  = 104407           848#   2087
RETURN  011660           925*    930    2041*   2043    2291#   2631
RSOS    011060          2131#   2677
RWRD0   012416           970     972    2579#
RWRD1   012420          1013    1015    2580#
RWRD10  012436          1314    1316    2587#
RWRD11  012440          1357    1359    2588#
RWRD12  012442          1400    1402    2589#
RWRD13  012444          1443    1445    2590#
RWRD14  012446          1486    1488    2591#
```

```
RWRD15  012450          1529    1531    2592#
RWRD16  012452          1572    1574    2593#
RWRD17  012454          1615    1617    2594#
RWRD2   012422          1056    1058    2581#
RWRD3   012424          1099    1101    2582#
RWRD4   012426          1142    1144    2583#
RWRD5   012430          1185    1187    2584#
RWRD6   012432          1228    1230    2585#
RWRD7   012434          1271    1273    2586#
SAVPC   011710          2101    2118*   2303#
SAVR0   011672          2127*   2131    2296#
SAVR1   011674          2126*   2132    2297#
SAVR2   011676          2125*   2133    2298#
SAVR3   011700          2124*   2134    2299#
SAVR4   011702          2123*   2135    2300#   2720
SAVR5   011704          2122*   2136    2301#   2718
SAVSP   011706          2302#   2605*   2612
SAVOSP= 104406           847#   2064
SCOPE = 104400           841#    974    1017    1060    1103    1146    1189    1232    1275    1318    1361    1404    1447
                        1490    1533    1576    1619    1643    1667    1691    1715    1739    1763    1787    1811    1835
                        1859    1883    1907    1931    1955    1979    2003
SCOPER  010446          2030#   2670
SCOPE1= 104410           849#
SCOP1R  010546          2051#   2678
SPACNT= 011605          2234*   2252    2255*   2265#
STACK = 013540           535#    864     913    2621
START   001000           833     863#
STFLG   011714           866*    926     928*   2305#
SVOS    011026          2122#
SVOSP   011020          2118#   2676
SWR   = 177570           532#    876     914    2030    2032    2034    2051    2058    2088    2095
SW00  = 000001           515#    876
SW01  = 000002           514#    914
SW02  = 000004           513#
SW03  = 000010           512#
SW04  = 000020           511#
SW05  = 000040           510#
SW06  = 000100           509#
SW08  = 000400           508#
SW09  = 001000           507#   2051
SW10  = 002000           506#   2030    2095
SW11  = 004000           505#   2034
SW12  = 010000           504#
SW13  = 020000           503#   2058
SW14  = 040000           502#   2032
SW15  = 100000           501#
TBUF    011754           956     999    1042    1085    1128    1171    1214    1257    1300    1343    1386    1429    1472
                        1515    1558    1601    1632    1656    1680    1704    1728    1752    1776    1800    1824    1848
                        1872    1896    1920    1944    1968    1992   2320#
TDAT  = 000400          2320#   2321    2322#   2323#   2324#   2325#   2326#   2327#   2328#   2329#   2330#   2331#   2332#
                        2333#   2334#   2335#   2336#   2337#   2338#   2339#   2340#   2341#   2342#   2343#   2344#   2345#
                        2346#   2347#   2348#   2349#   2350#   2351#   2352#   2353#   2354#   2355#   2356#   2357#   2358#
                        2359#   2360#   2361#   2362#   2363#   2364#   2365#   2366#   2367#   2368#   2369#   2370#   2371#
                        2372#   2373#   2374#   2375#   2376#   2377#   2378#   2379#   2380#   2381#   2382#   2383#   2384#
                        2385#   2386#   2387#   2388#   2389#   2390#   2391#   2392#   2393#   2394#   2395#   2396#   2397#
                        2398#   2399#   2400#   2401#   2402#   2403#   2404#   2405#   2406#   2407#   2408#   2409#   2410#
```

```
                    2411#  2412#  2413#  2414#  2415#  2416#  2417#  2418#  2419#  2420#  2421#  2422#  2423#
                    2424#  2425#  2426#  2427#  2428#  2429#  2430#  2431#  2432#  2433#  2434#  2435#  2436#
                    2437#  2438#  2439#  2440#  2441#  2442#  2443#  2444#  2445#  2446#  2447#  2448#  2449#
                    2450#  2451#  2452#  2453#  2454#  2455#  2456#  2457#  2458#  2459#  2460#  2461#  2462#
                    2463#  2464#  2465#  2466#  2467#  2468#  2469#  2470#  2471#  2472#  2473#  2474#  2475#
                    2476#  2477#  2478#  2479#  2480#  2481#  2482#  2483#  2484#  2485#  2486#  2487#  2488#
                    2489#  2490#  2491#  2492#  2493#  2494#  2495#  2496#  2497#  2498#  2499#  2500#  2501#
                    2502#  2503#  2504#  2505#  2506#  2507#  2508#  2509#  2510#  2511#  2512#  2513#  2514#
                    2515#  2516#  2517#  2518#  2519#  2520#  2521#  2522#  2523#  2524#  2525#  2526#  2527#
                    2528#  2529#  2530#  2531#  2532#  2533#  2534#  2535#  2536#  2537#  2538#  2539#  2540#
                    2541#  2542#  2543#  2544#  2545#  2546#  2547#  2548#  2549#  2550#  2551#  2552#  2553#
                    2554#  2555#  2556#  2557#  2558#  2559#  2560#  2561#  2562#  2563#  2564#  2565#  2566#
                    2567#  2568#  2569#  2570#  2571#  2572#  2573#  2574#  2575#  2576#
TEMP    013126      2238   2622*  2623*  2684#
TKCSR   011610      2159   2269#
TKDBR   011612      2161   2165   2270#
TPCSR   011614      2143   2166   2271#
TPDBR   011616      2148*  2165*  2272#
TRPOK   011000      2110#
TRPSRV  010766       830   2107#
TRPTAB  013072      2112   2670#
TYPDAT  010702      2077   2083#
TYPE  = 104401       842#   871    929   2012   2081   2155   2170   2228   2258   2627
TYPER   011112      2141#  2671
TYPMSG  010664      2075   2078#
T1      001274       925    940#
T10     003362      1241#
T11     003614      1284#
T12     004046      1327#
T13     004300      1370#
T14     004532      1413#
T15     004764      1456#
T16     005216      1499#
T17     005450      1542#
T2      001526       983#
T20     005702      1585#
T21     006134      1625#
T22     006246      1649#
T23     006360      1673#
T24     006472      1697#
T25     006604      1721#
T26     006716      1745#
T27     007030      1769#
T3      001760      1026#
T30     007142      1793#
T31     007254      1817#
T32     007366      1841#
T33     007500      1865#
T34     007612      1889#
T35     007724      1913#
T36     010036      1937#
T37     010150      1961#
T4      002212      1069#
T40     010262      1985#
T5      002444      1112#
T6      002676      1155#
```

```
T7          003130          1198#
VEC1        001060           873     876#
VEC2        001070           875     878#
WRDCNT      011602          2232*   2260*   2263#
X       =   000000             1#
XBIT    =   000000           932#    1620#
XLINE   =   000020           932#    1620#
XN      =   000041             1#     940     943#    983     986#   1026    1029#   1069    1072#   1112    1115#   1155    1158#
                            1198    1201#   1241    1244#   1284    1287#   1327    1330#   1370    1373#   1413    1416#   1456
                            1459#   1499    1502#   1542    1545#   1585    1588#   1625    1628#   1649    1652#   1673    1676#
                            1697    1700#   1721    1724#   1745    1748#   1769    1772#   1793    1796#   1817    1820#   1841
                            1944#   1865    1868#   1889    1892#   1913    1916#   1937    1940#   1961    1964#   1985    1988#
Y       =   000011             1#     841     842#    843#    844#    845#    846#    847#    848#    849#    850#
.       =   013342           565#    566     568     570     572     574     576     578     580     582     584     586     588
                             590     592     594     596     598     600     602     604     606     608     610     612     614
                             616     618     620     622     624     626     628     630     632     634     636     638     640
                             642     644     646     648     650     652     654     656     658     660     662     664     666
                             668     670     672     674     676     678     680     682     684     686     688     690     692
                             694     696     698     700     702     704     706     708     710     712     714     716     718
                             720     722     724     726     728     730     732     734     736     738     740     742     744
                             746     748     750     752     754     756     758     760     762     764     766     768     770
                             772     774     776     778     780     782     784     786     788     790     792     794     796
                             798     800     802     804     806     808     810     812     814     816     818     820     825#
                             932#    850#    852#    854#   2578#   2608    2624    2666#   2683#   2685#   2687#   2715#
```

```
BLIND1    931#   1620   1644   1668   1692   1716   1740   1764   1788   1812   1836   1860   1884   1908   1932
         1956   1980
BREAK1    931#    932    975   1018   1061   1104   1147   1190   1233   1276   1319   1362   1405   1448   1491
         1534   1577
CCLRAL    932#   2310
CODEMI    854#    903
HLT       545#    968    973   1011   1016   1054   1059   1097   1102   1140   1145   1183   1188   1226   1231
         1269   1274   1312   1317   1355   1360   1398   1403   1441   1446   1484   1489   1527   1532   1570
         1575   1613   1618   1642   1666   1690   1714   1738   1762   1786   1810   1834   1858   1882   1906
         1930   1954   1978   2002
SSETAL    931#
TRPDEF      1#    841    842    843    844    845    846    847    848    849
TS          1#    940    983   1026   1069   1112   1155   1198   1241   1284   1327   1370   1413   1456   1499
         1542   1585   1625   1649   1673   1697   1721   1745   1769   1793   1817   1841   1865   1889   1913
         1937   1961   1985
WORDS    2579#  2580   2581   2582   2583   2584   2585   2586   2587   2588   2589   2590   2591   2592   2593
         2594
.BUFFE      1#   2679
.CATCH      1#    564
.ENDCO      1#   2721
.EOP        1#   2005
.ERROR      1#   2055
.ERRTA      1#   2688
.HEADE      1#    479
.INSTR      1#   2150
.MSG        1#   2635
.OCTAS      1#   2225
.PARAM      1#   2174
.PFAIL      1#   2595
.POINT      1#   2267
.RESRE      1#   2129
.SAVRE      1#   2115
.SCOPE      1#   2026
.SCOPI      1#   2048
.SETVE      1#    822
.START      1#    854
.SYMBO      1#    498
.TRPDE      1#    836
.TRPSR      1#   2102
.TRPTA      1#   2667
.TYPER      1#   2138
.VARIA      1#   2286

. ABS.  013342     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZDHIC.BIN,DZDHIC.SEQ/CRF/SOL/DOC=UTIL2.P11,DZDHIC.P11
 RUN-TIME: 14 24 2 SECONDS
 RUN-TIME RATIO: 221/41=5.3
 CORE USED:  11K  (21 PAGES)

 DOCUMENT PAGES:  65
```

E06

Spooler runtime 11 Seconds, 49 KCS, 259 disk reads, 4 disk writes, 66 pages

[redacted] Date 03-Dec-76 09:52:09 Monitor IPC-D 602 [630] [redacted]