DM11A

LOGIC TEST

MD-11-DZDMA-B

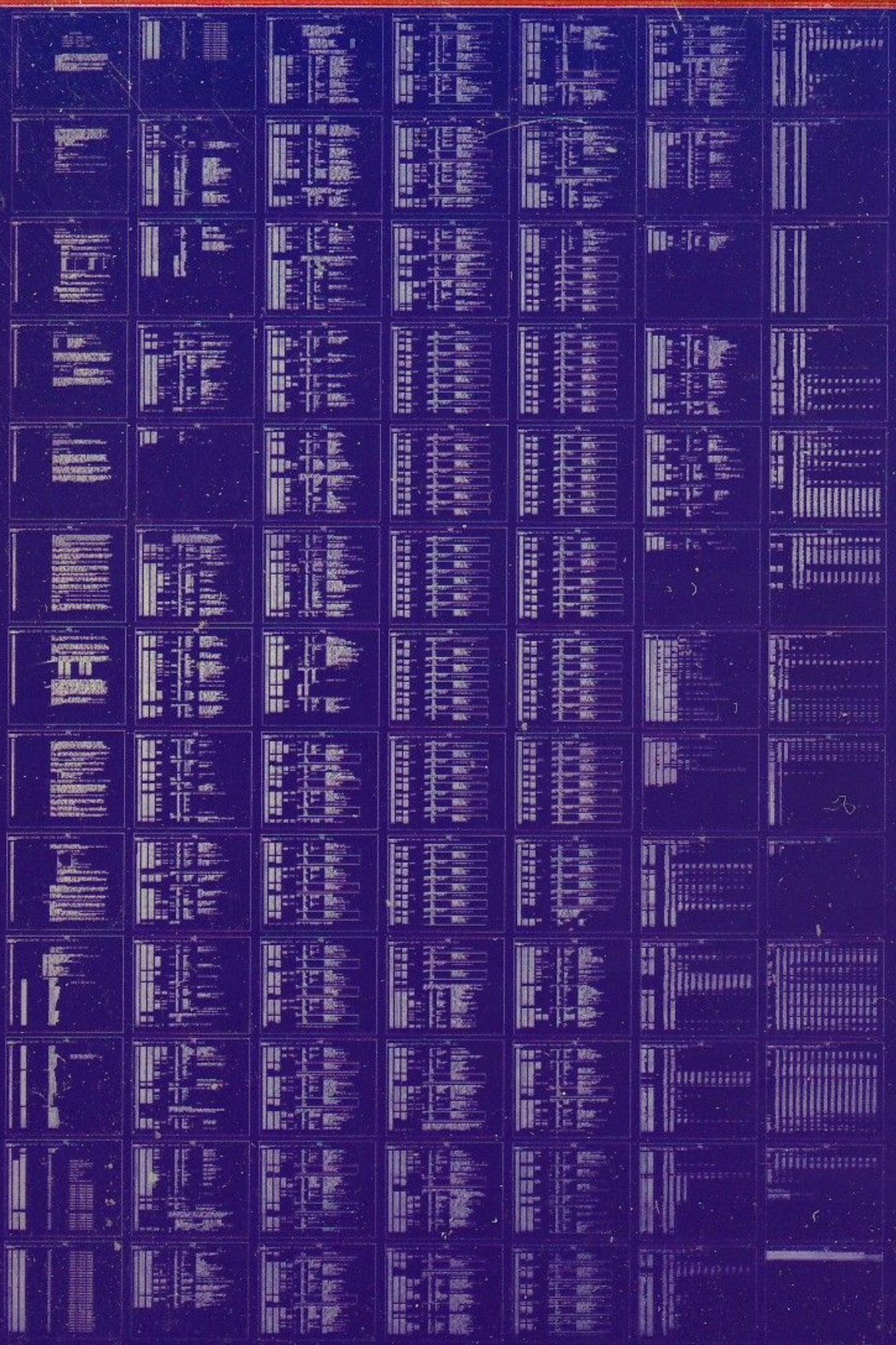
EP-DZDMA-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A



```
.REM %
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDMA-B-D

PRODUCT NAME: DM11 LOGIC TESTS

DATE RELEASED: MAY, 1976

MAINTAINER: DIAGNOSTIC GROUP

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR TESTING THE DM11 (ASYNCHRONOUS DATA MULTIPLEXER). MAINDEC-11-DZDMA (DM11 LOGIC TESTS), AND MAINDEC-11-DZDMB (DM11 MULTIPLE LINE DATA TESTS). THE LOGIC TESTS INDIVIDUALLY TEST EACH OF THE 16 DM11 LINES AND ALL COMMOM LOGIC. THE MULTIPLE LINE DATA TESTS RUN SEVERAL LINES CONCURRENTLY AND ARE USED TO TEST LINE INTERACTION AND DATA TRANSMISSION/RECEPTION RELIABILITY.
THIS DOCUMENT DESCRIBES THE LOGIC TESTS.
THE AVAILABLE TESTS ARE:
PRG0 - LOGIC TEST
PRG1 - TRANSMITTER SCOPE LOOP
PRG2 - TRANSMIT/RECEIVE SCOPE LOOP

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP 11 FAMILY PROCESSER
B. DM11
C. JUMPERS CONNECTING 16 TRANSMITTERS TO THEIR RESPECTIVE RECEIVERS.

2.2 STORAGE

THIS PROGRAM USES ALL OF CORE (8K) EXCEPT THAT AREA RESERVED FOR THE LOADERS.

3. LOADING PROCEEDURE

THE ABSOLUTE LOADER IS USED TO LOAD THE PROGRAM.

4. USE PROCEDURE

4.1 STARTING PROCEDURE

BEFORE STARTING MAKE SURE THAT THE TTY IS IN REMOTE MODE, AND THE JUMPERS ARE INSTALLED. THREE STARTING ADDRESSES ARE PROVIDED.

0200 - THIS STARTING ADDRESS REQUESTS DM11 PARAMETERS, AND MUST BE USED TO INITIALLY START THE PROGRAM, AND WHENEVER ANY OF THE PARAMETERS LISTED BELOW IS CHANGED.

A. VECTOR ADDRESS ?
RESPONSE: TYPE IN THE VECTOR ADDRESS OF THE DM11 RECEIVER UNDER TEST. CARRIAGE RETURN SELECTS 0300

B. UNIT #(8)?
RESPONSE: THE DM11 UNIT NUMBER CORRESPONDS TO THE ADDRESS TO WHICH THE CLOCK STATUS REGISTER (CSR) RESPONDS.

CSR ADDRESS	DM11 UNIT #	CSR ADDRESS	DM11 UNIT #
175000	0	175100	10
175010	1	175110	11
175020	2	175120	12
175030	3	175130	13
175040	4	175140	14
175050	5	175150	15
175060	6	175160	16
175070	7	175170	17

CARRIAGE RETURN SELECTS UNIT # 0.

C. WHAT IS THE CHARACTER LENGTH?
RESPONSE: CHARACTER LENGTH REFERS TO THE NUMBER OF DATA BITS PER CHARACTER (5-8). CARRIAGE RETURN SELECTS 8 DATA BITS PER CHARACTER.

D. PRG #
RESPONSE: TYPE PROGRAM NUMBER OF PROGRAM YOU WISH TO RUN. CARRIAGE RETURN SELECTS PROGRAM # 0.

NOTES:
CARRIAGE RETURN TERMINATES ALL RESPONSES.
ANY UNACCEPTABLE RESPONSE WILL RESULT IN A ? TYPEOUT AND THE PARAMETER WILL AGAIN BE REQUESTED.

0204 - THIS STARTING ADDRESS USES PREVIOUSLY DEFINED DM11 PARAMETERS AND REQUESTS THE PROGRAM NUMBER OF THE PROGRAM YOU WISH TO RUN.

0210 - THIS STARTING ADDRESS STARTS THE PREVIOUSLY SELECTED PROGRAM USING PREVIOUSLY SELECTED PARAMETERS.

4.2 SWITCH SETTINGS

THE FOLLOWING SWITCH SETTINGS APPLY TO PROGRAM #0.

SR 0-6	ROUTINE TO BE RUN (IF ENABLED BY SR-9)
SR 8	RING BELL ON ERROR
SR 9	LOOP SELECTED ROUTINE
SR 11	INHIBIT ITERATION (DO EACH ROUTINE ONCE)
SR 13	INHIBIT PRINTOUT
SR 14	SCOPE (LOOP ROUTINE)
SR 15	HALT ON ERROR

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT
A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS
THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A
SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO.
IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE
DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

```
(IE)    SWR=XXXXXX  NEW=
```

POSSIBLE RESPONSES ARE:

1.	<CR>	IF NO CHANGES ARE TO BE MADE
2.	6 DIGITS 0-7	TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE .LAST DIGIT FOLLOWED BY <CR>.
3.	↑U	TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE
CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G
(CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE
THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM
CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER
APPLICABLE AREAS.

```
174
175
176                             5.0     PROGRAM DESCRIPTION
177
178                             5.1     PRGO - LOGIC TESTS
179
180                                     PRGO CONSISTS OF 152(8) INDEPENDENT ROUTINES WHICH TEST
181                             VARIOUS FUNCTIONS OF THE DM11 HARDWARE.  ANY OF THESE ROUTINES
182                             MAY BE INDIVIDUALLY SELECTED AND RUN (SEE SEC. 4.2 FOR SWITCH
183                             SETTING)
184
185
186                             5.1.1   ROUTINE DESCRIPTION
187
188                                     ROUTINE         TESTS
189
190                                     RT0     TESTS THE ABILITY TO REFERENCE THE FOUR DM11 REGISTERS
191                             (CONTROL STATUS REGISTER (CSR), BUFFER ACTIVE REGISTER (BAR),
192                             BREAK STATUS REGISTER (BKCSR), AND THE BASE REGISTER (BASREG)).
193                             IF AN ILLEGAL REFERENCE OCCURS WHEN THE CSR IS REFERENCED THE
194                             PROGRAM WILL INDICATE AN ERROR, AND AUTOMATICALLY LOOP THE ERROR
195                             AS LONG AS THE ERROR CONDITION EXISTS.
196                             RT0     PC=XXXXXX
197
198                                     RT1-RT10 BIT 'BANGS' THE CSR (BITS 0,1,2,4,5,6,12,13), TESTING THAT
199                             EACH BIT IN THE CSR CAN BE INDIVIDUALLY SET AND CLEARED.  TWO ERROR
200                             TYPES ARE DETECTED IN THESE TESTS, A BIT FAILED TO SET, AND/OR A
201                             BIT FAILED TO CLEAR.  THE ERROR PRINTOUT SHOWS THE ROUTINE THAT
202                             FAILED AND THE PC WHERE THE ERROR WAS DETECTED.
203
204                                     RT11- TESTS THAT RESET AND CLEAR CLEAR ALL R/W BITS IN THE CSR.  TWO
205                             ERROR TYPES ARE DETECTED IN THIS ROUTINE SHOWING THE CONTENTS OF THE
206                             CSR AFTER THE RESET & CLEAR INSTRUCTION.  THE PROGRAM AUTOMATICALLY
207                             LOOPS IF AN ERROR OCCURS.  SHOWN BELOW IS THE ERROR TYPEOUT.
208                             RT11    PC=XXXXXX       ERR S/B:000000 WAS:XXXXXX
```

RT12 LOADS A BINARY COUNT PATTERN INTO THE BKCSR AND READS BACK THE RESULTS. IF THE DATA READ BACK IS INCORRECT AN ERROR IS INDICATED. THE SCOPE SWITCH WILL CAUSE THE PROGRAM TO RELOAD THE BINARY NUMBER AND REPEAT THE TEST. THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL RESULTS.
THE SECOND PORTION OF THE TEST CLEARS THE PREVIOUSLY LOADED NUMBER IF THE SCOPE SWITCH IS SET THE PROGRAM LOOPS BACK AND REPEATS THE CLEAR INSTRUCTION.

RT13 THIS ROUTINE LOADS RANDOM NUMBERS INTO THE BKCSR. IF A RANDOM NUMBER IS LOADED INCORRECTLY AN ERROR IS INDICATED SHOWING THE CORRECT AND ACTUAL RESULTS.

RT14 THIS ROUTINE TESTS THAT RESET WILL CLEAR ALL BREAK STATUS REGISTER (BKCSR) BITS. IF ALL BITS DO NOT CLEAR WHEN THE RESET IS GIVEN AN ERROR IS INDICATED. THE ERROR TYPEOUT SHOWS THE CORRECT RESULT (ALL 0'S) AND THE ACTUAL RESULT.

RT15-RT16 THESE ROUTINES ARE THE SAME AS RT12 & RT13 EXCEPT THAT THE BASE REGISTER IS TESTED.

RT17 THIS ROUTINE TESTS THAT ALL BAR BITS CAN BE INDIVIDUALLY SET AND CLEARED. THE ROUTINE SHIFTS A '1' THROUGH THE BAR THEREBY SETTING EACH BAR BIT AND THEN THE BAR BIT IS CLEARED. THE ERROR TYPEOUTS SHOW CORRECT AND ACTUAL RESULTS.

RT20 THIS ROUTINE TESTS THAT RESET AND CLEAR CLEAR ALL BAR BITS THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL RESULTS.

RT21-RT23 THESE ROUTINES TEST THAT THE CSR, BAR, AND BKCSR RESPOND PROPERLY TO BYTE COMMANDS. BOTH BYTES ARE REFERENCED IN THESE ROUTINES USING CLRB INSTRUCTIONS. THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL RESULTS.

RT24 THIS ROUTINE TESTS THAT THE DM11 CAN INTERRUPT THE PROCESSOR VIA THE OVER RUN BIT (CSR BIT 13). THE ERROR TYPEOUT SHOWS THE ROUTINE NUMBER AND THE PC WHERE THE ERROR WAS DETECTED.

RT25 THIS ROUTINE TESTS THAT THE DM11 INTERRUPTS THE PROCESSER AT THE PROPER LEVEL.

RT26-RT45 THESE ROUTINES TEST THE BASIC TRANSMITTER FUNCTIONS ON EACH LINE

RT46-RT65 THESE ROUTINES TEST THE BASIC RECEIVER FUNCTIONS ON EACH LINE

RT66 THIS ROUTINE TESTS THAT THE DM11 WILL SET THE NEX BIT (CSR BIT 14). WHEN THE DM11 TRIES TO TRANSMIT FROM NON-EXISTANT MEMORY. ALL LINES ARE INDIVIDUALLY TRANSMITTED ON. THE ERROR TYPEOUT SHOWS THE FAILING LINE. ALSO TESTED IS THAT THE NEX BIT WHEN SET CAUSES AN INTERRUPT.

```
                RT67    THIS ROUTINE TESTS THAT THE NEX BIT (CSR BIT 14) SETS
                WHEN THE DM11 TRIES TO REFERENCE THE TUMBLE TABLE THAT IS IN NON-
                EXISTANT MEMORY.


                RT70    THIS ROUTINE TESTS THAT WHEN THE GO BIT (CSR BIT 0) IS
                CLEAR THAT NO DATA IS RECEIVED ON ANY LINE.  ALL LINES ARE TRANS-
                MITTED ON AND AFTER THE TRANSMISSION IS COMPLETE THE RECEIVER DONE
                FLAG IS TESTED. THE ERROR TYPEOUT SHOWS THE LINE ON WHICH DATA WAS
                RECEIVED.
                THE TYPEOUT SHOWN BELOW SHOWS THAT DATA WAS RECEIVED ON LINE 0
                RT70    PC=XXXXXX       ERRS/B:000001 WAS: 000001

                RT71    THIS ROUTINE TESTS THAT THE CURRENT ADDRESS IS INCREMENTED
                PROPERLY BY THE DM11.  THE TABLE BELOW SHOWS THE ADDRESS LOADED
                INTO IN THE CURRENT ADDRESS TABLE BEFORE 2 CHARACTERS ARE TRANSMITTED
                THE RESULTANT ADDRESS AFTER THE CHARACTER IS TRANSMITTED
                        BEFORE  AFTER           BEFORE  AFTER
                        000000  000001          000777  001000
                        000001  000002          001777  002000
                        000003  000004          003777  004000
                        000007  000010          007777  010000
                        000017  000020          017777  020000
                        000037  000040          037777  040000
                        000077  000100          077777  100000
                        000177  000200          177777  000000

                        000377  000400

                THE ERROR TYPEOUT SHOWS CORRECT AND ACTUAL CURRENT ADDRESS.
                RT72    THIS ROUTINE TESTS THAT DATA CAN BE TRANSMITTED FROM ALL
                AVAILABLE CORE AND RECEIVED CORRECTLY.  THIS IS DONE BY TRANS-
                MITTING 1 CHARACTER FROM SEVERAL ADDRESSES IN EACH 4K BLOCK OF
                CORE ON LINE 0.  THE ERROR TYPEOUT WILL SHOW TRANSMITTED AND
                ACTUAL RECEIVED DATA.  IF A DATA ERROR RESULTED WHEN TRANSMITTING
                FROM THE FIRST 4K OF CORE EXAMINE THE CURRENT ADDRESS OF LINE 0
                TO DETERMINE WHERE IN THE FIRST 4K OF CORE THE DM11 WAS TRANSMIT-
                TING FROM WHEN ERROR OCCURRED, FOR ERRORS IN OTHER 4K BLOCKS THE
                CORRECT RESULT CORRELATES TO THE ADDRESS WHERE THE ERROR OCCUR-
                RED.  FOR EXAMPLE

                RT72 PC=XXXXXX      ERR S/B:000001  WAS XXXXXX.
                INDICATES THAT THE DM11 FAILED TO TRANSMIT AND RECEIVE CORRECT
                DATA WHEN TRANSMITTING FROM LOCATION 20000.
                THE TEST IS ABORTED BEFORE TRANSMITTING IF THE CORE LOCATION IS
                NON-EXISTANT.
```

RT73 THIS ROUTINE TESTS THAT THE TRANSMITTER CAN TRANSMIT 100 CHARACTERS ON EACH LINE. THE ROUTINE TESTS THAT EXACTLY 100 CHARACTERS HAVE BEEN TRANSMITTED BEFORE READY (CSR BIT15) SETS AND THE BAR BIT CLEARS. THE ERROR TYPEOUT GIVES THE NUMBER OF CHARACTERS RECEIVED AT THE TIME OF AN ERROR, AND THE FAILING LINE NUMBER (X2).

RT74 THIS ROUTINE TESTS THAT THE DM11 WILL STORE DATA SEQUENTIALLY IN THE TUMBLE TABLE AND ALSO THAT THE POINTER RETURNS TO THE TOP OF THE TABLE WHEN 64 CHARACTERS HAVE BEEN RECEIVED.

RT75-114 THESE ROUTINES CHECK THAT A BREAK CAN BE TRANSMITTED AND RECEIVED ON ALL ALINES.

R115-R134 THESE ROUTINES INDIVIDUALLY TRANSMIT, RECEIVE AND CHECK DATA PLUS PARITY ON EACH OF THE 16 DM11 LINES. ONLY DATA AND PARITY ERRORS ARE REPORTED.

RT131 THIS ROUTINE SIMULTANEOUSLY TRANSMITS AND RECEIVES A CHARACTER (ALL 1'S) ON THE 16 DM11 LINES. THE FOLLOWING TESTS ARE PERFORMED:

- A: THERE ARE 16 DATA ENTRIES (1 PER LINE)
- B: THERE ISN'T A 17TH ENTRY
- C: DATA IS CORRECT
- D: ONE ENTRY FOR EACH LINE

RT136 THIS ROUTINE TRANSMITS A BREAK ON EACH LINE. TESTS PERFORMED ARE THE SAME AS IN RT135.

RT137-RT144 THESE ROUTINES TRANSMIT 64 CHARACTERS ON EACH LINE WITH A DELAY BEFORE BEGINNING TRANSMISSION ON THE NEXT SUCCESSIVE LINE. THE DELAY BEFORE TRANSMITTING ON THE NEXT LINE IS HALVED BY SUCCESSIVE TESTS. NO DATA CHECKING IS PERFORMED BY THESE TESTS. TESTED ARE THAT OVER RUN (CSR BIT13) AND NEX (CSR BIT14) ARE NOT SET DURING TRANSMISSION/RECEPTION.

RT145 THIS ROUTINE TESTS PROPER OPERATION OF THE HALF DUPLEX BIT (CSR BIT1)

RT146 THIS ROUTINE TESTS THAT THE DM11 COMES TO AN 'ORDERLY HALT' WHEN THE RESET INSTRUCTION IS GIVEN. 'ORDERLY HALT' IS DEFINED AS CSR, BAR, AND BKCS CLEAR IMMEDIATLY AFTER THE RESET INSTRUCTION AND STAY CLEARED.

5.2 PRG1- TRANSMITTER SCOPE LOOP
PROGRAM 1 ALLOWS THE USER TO SCOPE THE DM11 TRANSMITTER FUNCTIONS WITH THE DM11 CONTINUOSLY RUNNING UNDER USER SUPPLIED PARAMETERS.

5.3 PRG2- TRANSMITTER/RECEIVER SCOPE LOOP
PROGRAM 2 ALLOWS THE USER TO SCOPE THE DM11 RECEIVER FUNCTIONS WITH THE DM11 CONTINUOULY RUNNING UNDER USER SUPPLIED PARAMETERS.

```
                        6.0     PROGRAM 1 AND PROGRAM 2 PARAMETERS
                                WHEN PROGRAM 1 OR PROGRAM 2 ARE SELECTED ADDITIONAL PARAMETERS WILL
                                BE REQUESTED BY EACH PROGRAM AS SHOWN BELOW.

                                        A: TYPE LINES TO BE TESTED
                                        EXAMPLES:
                                        TYPE            TO SELECT LINE(S)
                                        1               0
                                        3               1,0
                                        10              3
                                        17              3,2,1,0
                                        50              5,3
                                        3101            10,7,6,0
                                        17770           14,13,12,11,10,7,6,5,4,3
                                        177777          ALL
                                        NOTE, LINE NUMBERS ARE GIVEN IN OCTAL.

                                        B: HOW MANY CHARACTERS
                                        TYPE THE NUMBER OF CHARACTERS YOU WISH TO TRANSMIT. NOTE,
                                        THE NUMBER OF CHARACTERS MUST BE LESS THAN 200, AND IS TAKEN
                                        IN OCTAL.

                                        C: PUT CHARACTER IN SR (0-7);DELAY IN SR (8-15)
                                        SELF-EXPLANATORY. NOTE, THE DELAY REFERS TO A DELAY AFTER
                                        ALL THE CHARACTERS HAVE BEEN TRANSMITTED AND BEFORE A NEW
                                        TRANSMISSION PERIOD BEGINS.


                        7.0     PROGRAM LIMITATIONS
                                BECAUSE THE DM11 DIAGNOSTICS ARE INSENSITIVE TO 'REAL' ELAPSED TIME
                                THE DIAGNOSTIC DOES NOT 'KNOW' IF THE DM11 IS OPERATING AT THE COR-
                                RECT FREQUENCY OR THAT THE STOP CODE SELECTION LOGIC IS CORRECT,
                                THESE SHOULD BE CHECKED WITH A SCOPE.

                        8.0     PROGRAM NOTES
                                IF THE POWER FAILS THE PROGRAM TYPES AN ERROR MESSAGE INDICATING THE
                                ROUTINE THAT WAS RUNNING (PROG #0 ONLY) AND RESTARTS THE PROGRAM.

                                        ********** IMPORTANT NOTE **********
                        POWER FAIL TEST
                                A TEST OF THE POWER FAIL LOGIC SHOULD BE PERFORMED ON EACH UNIT.
                                SELECT & RUN ROUTINE 144 (L.A. = 210 SR =5144 PRESS START). TURN
                                THE POWER OFF THEN ON. THE PROGRAM WILL TYPE OUT THE POWER FAIL
                                ERROR
                                R144 PC=003622
                                AND CONTINUE RUNNING ROUTINE 144. LOWER SR 9 AND WAIT FOR END OF
                                TEST MESSAGE. 'TEST DZDMA COMPLETE'.
                                NOTE: IF THE POWER IS TURNED OFF DURING A RESET INSTRUCTION THE
                                PROGRAM WILL HALT. PRESS CONTINUE AND REPEAT THE TEST

                                IF THE PROGRAM HANGS THE BUS EXAMINE THE CONTENTS OF RTNNO. THE
                                CONTENTS OF RTNNO IS THE ROUTINE NUMBER THAT WAS RUNNING AT THE TIME
                                OF THE FAILURE.
```

```
                                %
                                        .TITLE  MAINDEC-11-DZDMA-B  DM11 LOGIC TESTS
                                        .NLIST  MC,MD,CND
                                        .LIST   ME
                                        .ENABLE ABS,AMA

                                ;DM11 LOGIC TESTS DIAGNOSTIC (MAINDEC-11-DZDMA)
                                ;COPYRIGHT 1972,1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                                ;PRG0- INPUT-OUTPUT LOGIC TESTS
                                ;PRG1- TRANSMITTER SCOPE LOOP
                                ;PRG2- TRANSMIT/RECEIVE SCOPE LOOP
                                ;STANDARD SR SWITCH OPTIONS (SWITCH SET TO A 1 )
                                ;SR15- HALT ON ERROR
                                ;SR14- SCOPE.
                                ;SR13- INHIBIT PRINTOUT
                                ;SR12- INHIBIT TRACE (NOT USED)
                                ;SR11- INHIBIT ITERATION.
                                ;SR10- LOOP PROGRAM. (NOT USED)
                                ;SR9- LOOP ROUTINE.
                                ;SR8-   RING BELL ON AN ERROR
                                ;SR6 THROUGH SR0 - NUMBER OF ROUTINE TO BE LOOPED.

                                ;EQUATE STATEMENTS
                                        CC=177776
                                        PSW=177776
                                        ERRVEC=4                ;ADDRESS OF ERROR TRAP VECTOR
                                        NOP=240
                                        OPEN=0
                                        MANUAL=BIT15
                                        LBIT17=100000
                                        LBIT16=40000
                                        LBIT15=20000
                                        LBIT14=10000
                                        LBIT13=4000
                                        LBIT12=2000
                                        LBIT11=1000
                                        LBIT10=400
                                        LBIT7=200
                                        LBIT6=100
                                        LBIT5=40
                                        LBIT4=20
                                        LBIT3=10
                                        LBIT2=4
                                        LBIT1=2
                                        LBIT0=1
                                        BIT15=100000
                                        BIT14=40000
                                        BIT13=20000
                                        BIT12=10000
                                        BIT11=4000
                                        BIT10=2000
                                        BIT9=1000
                                        BIT8=400
                                        BIT7=200
                                        BIT6=100
```

```
                000040                  BIT5=40
                000020                  BIT4=20
                000010                  BIT3=10
                000004                  BIT2=4
                000002                  BIT1=2
                000001                  BIT0=1
                005726                  POPSP=5726              ;POP THE STACK. SAME AS TST (6)+
                022626                  POPSP2=022626           ;POP STACK TWICE. SAME AS CMP (6)+,(6)+
                000340                  PRTY7=340               ;PRIORITY LEVEL DEFINITIONS
                000300                  PRTY6=300
                000240                  PRTY5=240
                000200                  PRTY4=200
                000140                  PRTY3=140
                000100                  PRTY2=100
                000040                  PRTY1=40
                000000                  PRTY0=0
                                ;LINE NUMBERS
                000000                  LINE0=0
                000002                  LINE1=2
                000004                  LINE2=4
                000006                  LINE3=6
                000010                  LINE4=10
                000012                  LINE5=12
                000014                  LINE6=14
                000016                  LINE7=16
                000020                  LINE10=20
                000022                  LINE11=22
                000024                  LINE12=24
                000026                  LINE13=26
                000030                  LINE14=30
                000032                  LINE15=32
                000034                  LINE16=34
                000036                  LINE17=36
                000000                  R0=%0
                000001                  R1=%1
                000002                  R2=%2
                000003                  R3=%3
                000004                  R4=%4
                000005                  R5=%5
                000006                  SP=%6
                000007                  PC=%7
                                ;EMT CALLS
                104000                  TYPE=EMT+0
                104001                  ERROR=EMT+1
                104002                  DATCHK=EMT+2
                104003                  CHALT=EMT+3
                104004                  EHALT=EMT+4
                104005                  SRESET=EMT+5
                104006                  SCOPE=EMT+6
                104007                  SAVREG=EMT+7
                104010                  RSTREG=EMT+10
                104011                  ERROR1=EMT+11
                104012                  INITIALIZE=EMT+12
                104013                  SUSWR=EMT+13
                104014                  KBDIN=EMT+14
                104015                  CNTLU=EMT+15
```

```
                    104400                  DELAY=TRAP+0
                    000007                  BELL=007
                    177777                  RTLAST=-1
                    000000                  Y=0
                    177777                  X=-1
                    000000                  A=0
                    000000                  .=0
000000  000002              .+2             ;UNASSIGNED TRAP
000002  000000              HALT
000004  000006      MACHER: .+2             ;SP OVERFLOW, BUS ERROR TRAP
000006  000000              HALT
000010  000012              .+2             ;RESERVED INSTRUCTION TRAP
000012  000000              HALT
000014  000016              .+2             ;TRACE TRAP
000016  000000              HALT
000020  000022              .+2             ;TRAP TO CALL IOX
000022  000000              HALT
000024  000026              .+2             ;POWER FAIL TRAP
000026  000000              HALT
000030  002542              EMTINT          ;EMT TRAP
000032  000340              PRTY7
000034  005264              DLY             ;TRAP TRAP. SIMILAR TO EMT
000036  000340              PRTY7
000040  000042              .+2
000042  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000044  000046              .+2
000046  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000050  000052              .+2
000052  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000054  000056              .+2
000056  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000060  000062              .+2
000062  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000064  000066              .+2
000066  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000070  000072              .+2
000072  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000074  000076              .+2
000076  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000100  000102              .+2
000102  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.


000104  000106              .+2
000106  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000110  000112              .+2
000112  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000114  000116              .+2
000116  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000120  000122              .+2
000122  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000124  000126              .+2
000126  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
000130  000132              .+2
000132  000000              HALT            ;TRAPPED TO PREVIOUS ADDRESS.
```

```
  585  000134  000136                .+2
  586  000136  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  587  000140  000142                .+2
  588  000142  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  589  000144  000146                .+2
  590  000146  000000                HALT        ;TRAPPED TO PREVICUS ADDRESS.
  591  000150  000152                .+2
  592  000152  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  593  000154  000156                .+2
  594  000156  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  595  000160  000162                .+2
  596  000162  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  597  000164  000166                .+2
  598  000166  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  599  000170  000172                .+2
  600  000172  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  601  000174  000176                .+2
  602  000176  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  603  000200  000202                .+2
  604  000202  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  605  000204  000206                .+2
  606  000206  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  607  000210  000212                .+2
  608  000212  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  609  000214  000216                .+2
  610  000216  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  611  000220  000222                .+2
  612  000222  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  613  000224  000226                .+2
  614  000226  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  615  000230  000232                .+2
  616  000232  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  617  000234  000236                .+2
  618  000236  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  619  000240  000242                .+2
  620  000242  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  621  000244  000246                .+2
  622  000246  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  623  000250  000252                .+2
  624  000252  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  625  000254  000256                .+2
  626  000256  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  627  000260  000262                .+2
  628  000262  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  629  000264  000266                .+2
  630  000266  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  631  000270  000272                .+2
  632  000272  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  633  000274  000276                .+2
  634  000276  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  635  000300  000302                .+2
  636  000302  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  637  000304  000306                .+2
  638  000306  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
  639  000310  000312                .+2
  640  000312  000000                HALT        ;TRAPPED TO PREVIOUS ADDRESS.
```

```
641  000314  000316                .+2
642  000316  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
643  000320  000322                .+2
644  000322  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
645  000324  000326                .+2
646  000326  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
647  000330  000332                .+2
648  000332  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
649  000334  000336                .+2
650  000336  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
651  000340  000342                .+2
652  000342  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
653  000344  000346                .+2
654  000346  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
655  000350  000352                .+2
656  000352  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
657  000354  000356                .+2
658  000356  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
659  000360  000362                .+2
660  000362  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
661  000364  000366                .+2
662  000366  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
663  000370  000372                .+2
664  000372  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
665  000374  000376                .+2
666  000376  000000                HALT            ;TRAPPED TO PREVIOUS ADDRESS.
667
```

```
                        000046                          .=46
000046  002500                                          $ENDAD
                        000052                          .=52
000052  060000                                          60000


                        000174                          .=174
000174  000000                          DISPREG:0
000176  000000                          SWREG:  0

                        000200                          .=200
000200  000137  002062                                  JMP     @#START         ;GO TO START OF DIAGNOSTIC.
000204  000137  002130                                  JMP     @#RSTAT1                ;GO GET PROGRAM # & RESTART PROGRAM
                                                                                ;USING PREVIOUS DM11 PARAMETERS
000210  000137  002206                                  JMP     @#RSTAT2                ;RESTART PREVIOUS PROGRAM USING
                                                                                ;PREVIOUS DM11 PARAMETERS


                        001100                          .=1100
001100  000000                          SPBOT:  0

001102  177570                          SWR:    177570
001104  177570                          DISPLAY:177570
001106  000000                          CAT:    OPEN                    ;STARTING ADDRESS OF
                        001146                          .=CAT+32.       ;CURRENT ADDRESS TABLE
001146  000000                          WCT:    OPEN                    ;STARTING ADDRES OF
                        001206                          .=WCT+32.       ;WORD COUNT TABLE
001206  000000                          BAT:    OPEN                    ;STARTING ADDRESS OF
                        001246                          .=BAT+32.       ;BIT ASSEMBLY TABLE
001246  000000                          VAC:    OPEN                    ;32. SPARE WORDS
001250  175000                          CSR:    175000                  ;ADDRESS OF CLOCK STATUS REGISTER
001252  175002                          BAR:    175002                  ;ADDRESS OF BUFFER ACTIVE REGISTER
001254  175004                          BRKCSR: 175004                  ;ADDRESS OF BREAK STATUS REGISTER
001256  175006                          BASREG: 175006                  ;ADDRESS OF BASE REGISTER
001260  000000                          CLKINT: OPEN                    ;DM11 VECTOR ADDRESS (RECEIVER)
001262  000240                          CLKLVL: PRTY5                   ;PRIORITY LEVEL
001264  000000                          XMTINT: OPEN                    ;DM11 VECTOR ADDRESS (TRANSMITTER)
001266  000240                          XMTLVL: PRTY5                   ;TRANSMITTER PRIORITY LEVEL
001270  000000                          TTDAT:  OPEN                    ;TUMBLE TABLE DATA
001272  000000                          LINBIT: OPEN                    ;LINE BIT (FOR BAR)
001274  000000                          BARDAT: OPEN                    ;BAR DATA
001276  000000                          TTPTR:  OPEN                    ;PROGRAM TUMBLE TABLE POINTER
001300  000000                          RCVDAT: OPEN
001302  000000                          XMTDAT: OPEN
001304  000000                          CARMSK: OPEN
                        001306                          .=VAC+32.
001306  000000                          TUMTAB: OPEN                    ;STARTING ADDRESS OF
                        001506                          .=TUMTAB+128.   ;TUMBLE TABLE
001506  000000                          KSTART: OPEN                    ;CURRENT PROGRAM START ADDRESS.
001510  000000                          CURTST: OPEN                    ;CONTAINS ADDR OF CURRENT TEST.
001512  000000                          RTNNO:  OPEN                    ;CONTAINS CURRENT TEST #.
001514  000000                          NXTST:  OPEN                    ;CONTAINS ADDR OF NEXT TEST.
001516  000000                          ICTR:   OPEN                    ;CONTAINS CURRENT ITERATION COUNT
001520  000000                          SCOPTR: OPEN                    ;CONTAINS CURRENT SCOPE POINTER.
001522  000000                          PRGLIM: OPEN
001524  006362                          PRGTAB: PRG0                    ;PRG0 START ADDRESS
```

```
  724  001526  016002                        PRG1            ;PRG1 START ADDRESS
  725  001530  016052                        PRG2            ;PRG2 START ADDRESS
  726  001532  006404              RSTART:   PRGOR           ;PRG0 RESTART ADDRESS
  727  001534  016014                        PRG1R           ;PRG1    "      "
  728  001536  016064                        PRG2R           ;PRG2    "      "
  729  001540  002764              EMTTAB:   TYP             ;POINTER TO TYPEOUT ROUTINE
  730  001542  001660                        ERR             ;POINTER TO ERROR ROUTINE
  731  001544  001640                        DTCHK           ;POINTER TO DATA COMPARISON ROUTINE
  732  001546  000000                        0
  733  001550  000000                        0
  734  001552  002674                        SRSETT          ;POINTER TO RESET ROUTINE
  735  001554  002364                        ESCOPE          ;POINTER TO SCOPE ROUTINE
  736  001556  002574                        SAVRG           ;POINTER TO SAVE REGISTERS ROUTINE
  737  001560  002634                        RSTRG           ;POINTER TO RESTORE REGISTERS ROUTINE
  738  001562  001676                        ERR1            ;POINTER TO ERROR1 ROUTINE
  739  001564  003076                        INIT            ;POINTER TO INITIALIZE ROUTINE
  740  001566  016660                        SUSWRR
  741  001570  016740                        KBDINTT
  742  001572  017004                        CNTLUU
  743  001574  177560              TKCSR:    177560
  744  001576  177562              TKDBR:    177562
  745  001600  177564              TPCSR:    177564
  746  001602  177566              TPDBR:    177566
  747  001604  000000              COUNT:    OPEN
  748  001606  000000              BFRPTR:   OPEN
  749  001610  000000              PARBIT:   OPEN
  750  001612  000000              PCADD:    OPEN
  751  001614  000000              APCADD:   OPEN
  752  001616  000000              PRVCNT:   OPEN
  753  001620  000000              TIME:     OPEN
  754  001622  175001              .CSR:     175001
  755  001624  175003              .BAR:     175003
  756  001626  175005              .BKCSR:   175005
  757  001630  175007              .BASREG:  175007
```

```
                                        ;ROUTINE TO TYPE OUT INCORRECT ROUTINE SELECTED
001632  104000                  INCRTN: TYPE
001634  017241                          MI                      ;TYPE INCORRECT ROUTINE SELECTED.
001636  000207                          RTS     %7              ;EXIT.

                                        ;DATA COMPARISON ROUTINE.
001640  123737  001300  001302  DTCHK:  CMPB    RCVDAT,XMTDAT   ;COMPARE RECEIVED & TRANSMITTED DATA
001646  001403                          BEQ     1$              ;CHARS. BRANCH IF SAME.
001650  004737  002034                  JSR     7,CNVDAT        ;CONVERT RCVDAT & XMTDAT TO ASCII
001654  104011                          ERROR1
001656  000002                  1$:     RTI                     ; EXIT.

                                        ;ERROR ROUTINE WHENEVER THE PROGAM DETECTS AN ERROR THE ERROR
                                        ;AND ERROR1 EMT INSTRUCTIONS ENTER HERE. ERROR AT ERR:,AND
                                        ;ERROR1 AT ERR1:
001660  012737  000402  001760  ERR:    MOV     #402,ERRB       ;MOV BR .+6 TO ERRB
001666  013737  001612  001614          MOV     PCADD,APCADD    ;GET PC WHERE ERROR OCCURRED
001674  000410                          BR      ERRA
001676  012737  000240  001760  ERR1:   MOV     #240,ERRB       ;MOVE NOP TO ERRB
001704  013737  001612  001614          MOV     PCADD,APCADD    ;GET PC WHERE ERROR OCCURRED
001712  004737  002034                  JSR     7,CNVDAT        ;CONVERT RCVDAT & XMIT DAT TO ASCII
001716  104014                  ERRA:   KBDIN                   ;GO CHECK FOR ↑G
001720  032777  020000  177154          BIT     #BIT13,@SWR     ;ERROR PRINTOUT DESIRED
001726  001017                          BNE     ERRC            ;BRANCH IF NO PRINTOUT
001730  004537  005440                  JSR     5,OACNV         ;CONVERT
001734  001614                          APCADD                  ;DATA
001736  017623                          APC                     ;TO
001740  000006                          6                       ;ASCII
001742  004537  005440                  JSR     5,OACNV         ;FOR
001746  001512                          RTNNO                   ;PRINTOUT
001750  017614                          ATNUMB
001752  000003                          3
001754  104000                          TYPE                    ;TYPE ERROR
001756  017611                          EMO                     ;MESSAGE
001760  000000                  ERRB:   OPEN                    ;NOP IF ERROR1, BR .+6 IF ERROR
001762  104000                          TYPE                    ;TYPE ANOTHER MESSAGE
001764  017173                          ERDAT                   ;IF ERROR 1
001766  032777  000400  177106  ERRC:   BIT     #BIT8,@SWR      ;RING BELL ON ERROR?
001774  001411                          BEQ     ERRD            ;BRANCH IF NO BELL ON ERROR
001776  105777  177576                  TSTB    @TPCSR          ;TELEPRINTER
002002  100375                          BPL     .-4             ;READY?
002004  012777  000007  177570          MOV     #BELL,@TPDBR    ;RING THE BELL
002012  105777  177562                  TSTB    @TPCSR          ;WAIT FOR THE BELL TO RING
002016  100375                          BPL     .-4
002020  005777  177056          ERRD:   TST     @SWR            ;HALT ON ERROR
002024  100001                          BPL     ERREX           ;GO TO EXIT IF NO HALT ON ERROR
002026  000000                          HALT
002030  104014                  ERREX:  KBDIN                   ;CHECK FOR  ↑G
002032  000002                          RTI                     ;RETURN

                                        ;SUBROUTINE TO CONVERT RCVDAT AND XMTDAT TO ASCII AND PLACE
                                        ;IN MESSAGE.
002034  004537  005440          CNVDAT: JSR     5,OACNV
```

```
814   002040   001302                        XMTDAT
815   002042   017205                        AASB
816   002044   000006                        6
817   002046   004537   005440               JSR     5,OACNV
818   002052   001300                        RCVDAT
819   002054   017221                        AWAS
820   002056   000006                        6
821   002060   000207                        RTS     7               ;EXIT
822
823
```

```
                                        ;THE FIRST PART OF THE START ROUTINE CONTAINS A SHORT
                                        ;ROUTINE TO CHECK FOR MEMORY MANAGEMENT, ALTHOUGH THIS
                                        ;DIAGNOSTIC DOES NOT USE MEMORY MANAGEMENT, ITS PRESENCE
                                        ;INDICATES THAT OVER 28K OF MEMORY MAY BE PRESENT IN WHICH
                                        ;CASE TESTS RT66 AND RT67 MAY FAIL. IF MEM. MAN. IS
                                        ;PRESENT THESE TESTS ARE SKIPPED BY THE PROGRAM.

002062  012706  001100        START:  MOV     #SPBOT,%6
002066  104013                        SUSWR                   ;SEE IF SWITCH-LESS PROCESSOR
002070  012737  002112  000004        MOV     #1$,@#ERRVEC    ;SET UP FOR ERROR TRAP
002076  005737  172300                TST     @#172300        ;TEST FOR KT11
002102  012737  012202  011614        MOV     #RT70,@#RT65+2  ;KT11 PRESENT, SET UP TO
                                                              ;SKIP RT66 AND RT67
002110  000402                        BR      .+6
002112  012706  001100        1$:     MOV     #SPBOT,%6       ;TRAP OCCURRED, NO KT11 PRESENT,
                                                              ;RESET STACK
002116  012737  000006  000004        MOV     #ERRVEC+2,@#ERRVEC      ;RESET ERROR TRAP


                                                              ;OUT INTERRUPTS (SET PRIORITY LEVEL 7)
002124  004737  003126                JSR     7,@#DMPAR               ;GET DM11 PARAMETERS
002130  012706  001100        RSTAT1: MOV     #SPBOT,%6
002134  104012                        INITIALIZE
002136  104000                        TYPE
002140  017230                        M0
002142  004537  003702                JSR     5,RECD          ;GET PRGNUM AND PUT IT
002146  000000        PRGNUM: 0               ;HERE
002150  043737  001522  002146        BIC     PRGLIM,PRGNUM   ;MASK OFF UNUSED BITS
002156  006337  002146                ASL     PRGNUM          ;SHIFT PROGRAM #
002162  012737  004062  000024        MOV     #PFAIL,24
002170  012737  000340  000026        MOV     #PRTY7,26
002176  013700  002146                MOV     PRGNUM,%0       ;GET PROGRAM #
002202  000170  001524                JMP     @PRGTAB(0)      ;GO START PROGRAM
002206  012737  004062  000024        MOV     #PFAIL,24
002214  012737  000340  000026        MOV     #PRTY7,26
002222  012706  001100                MOV     #SPBOT,%6
002226  104012                        INITIALIZE
002230  013700  002146                MOV     PRGNUM,%0       ;GET PROGRAM #
002234  000170  001532                JMP     @RSTART(0)      ;GO RESTART PROGRAM
002240  022737  000176  001102        CMP     #SWREG,SWR
002246  001404                        BEQ     1$
002250  104000                        TYPE                    ;TYPE MESSAGE TO REQUEST SWITCH
002252  017271                        M3                      ;REGISTER SETTINGS
002254  000000                        HALT                    ;WAIT FOR OPERATOR TO SET SWITCHES
002256  000401                        BR      GETRDY
002260  104015          1$:           CNTLU                   ;GO GET SWREG SETTINGS
002264  013737  001506  001514        MOV     KSTART,NXTST    ;ADDR OF 1ST ROUTINE TO NXTST
002270  012737  000006  000004        MOV     #6,@#ERRVEC             ;RESET ERROR TRAP VECTOR
002276  000240                        NOP
002300  104012                        INITIALIZE
002302  004737  002510                JSR     %7,FORWD        ;ROLL FORWARD TO "NEXT" ROUTINE.
002306  032777  001000  176566        BIT     #BIT9,@SWR      ;CHECK SELECT ROUTINE SWITCH
002314  001003                        BNE     GTRDYC          ;BRANCH IF SELECT ROUTINE SWITCH IS SET.
002316  000177  177166                JMP     @CURTST         ;GO RUN CURRENT ROUTINE.
002322  000450                        BR      SCOPED          ;NO GO. MANUAL RTN BYPASSED.
002324  017700  176552        GTRDYC: MOV     @SWR,%0         ;(SR) TO R0
```

```
                                                    BIC     #177600,%0      ;MASK UNDESIRED BITS
                                                    CMPB    RTNNO,%0        ;COMPARE RTNNO TO (R0)
                                                    BNE     GTRDY0          ;BRANCH IF ROUTINE NOT FOUND YET.
                                                    JMP     @CURTST         ;GO RUN ROUTINE.
001514                                  GTRDYD:     CMP     #-1,NXTST       ;NO. CHECK FOR LAST ROUTINE.
                                                    BNE     GTRDYA          ;BRANCH IF NOT LAST ROUTINE.
001632                                              JSR     %7,INCRTN       ;YES, INCORRECT ROUTINE SELECTED.
                                                    BR      GETRDY          ;START OVER.

                                        ;SCOPE SERVICE ROUTINE
                                        ESCOPE:     NOP
                                                    KBDIN                   ;CHECK FOR ↑G
                                                    CLR     @BAR            ;CLEAR ALL DM11 REGISTERS
                                                    CLR     @CSR            ;AND SET BASE REGISTER
                                                    CLR     @BKCSR          ;AT THE STARTING ADDRESS
                                                    INITIALIZE
001520                                              MOV     SCOPTR,(SP)
176462                                              BIT     #BIT14,@SWR     ;CHECK FOR SCOPE OPTION.
                                                    BEQ     SCOPEB          ;BRANCH IF SCOPE SW NOT SET.
000000                                  SCOPEA:     JMP     @(SP)           ;RETURN TO ROUTINE
176446                                  SCOPEB:     BIT     #BIT11,@SWR     ;TEST INHIBIT ITERATION SWITCH
                                                    BNE     SCOPED          ;BRANCH IF INHIBIT ITERATION SW SET.
001516                                              DEC     ICTR            ;DECREMENT ITERATION COUNT.
                                                    BNE     SCOPEA          ;BRANCH IF COUNT NOT 0.
176430                                  SCOPED:     BIT     #BIT9,@SWR      ;CHECK SELECT ROUTINE SWITCH
                                                    BNE     GETRDY          ;BRANCH IF SELECT RTN SW SET
001514                                              CMP     #-1,NXTST       ;LAST TEST?
                                                    BNE     GTRDYX          ;BRANCH IF NOT LAST TEST.
                                                    TYPE                    ;TYPE
                                                    M2                      ;'PRGEND'
000042                                              MOV     @#42,%0         ;CHECK XXDP/ACT11 MONITOR HOOK
                                                    BEQ     GETRDY
                                                    RESET
                                        SENDAD:     JSR     7,(0)           ;RETURN TO XXDP/ACT11 MONITOR
                                                    NOP
                                                    NOP
                                                    NOP
001514                                  FORWD:      MOV     NXTST,%5        ;ADDR OF NEXT ROUTINE TO R5.
001512                                              MOV     (5)+,RTNNO      ;GET NEXT ROUTINE NUMBER.
001514                                              MOV     (5)+,NXTST      ;GET ADDR OF NEXT "NEXT" ROUTINE.
001516                                              MOV     (5)+,ICTR       ;GET ITERATION COUNT.
001520                                              MOV     (5)+,SCOPTR     ;GET SCOPE LOOP ENTRY POINTER.
001510                                  FORWDA:     MOV     %5,CURTST       ;ADDR OF NOW CURRENT TEST TO CURTST.
                                                    RTS     %7              ;EXIT FORWD SUBROUTINE.


                                        ;EMT TRAP INTERPRETER
                                        EMTINT:     MOV     (6)+,-(6)       ;GET PC OF NEXT INSTRUCTION
000002                                              SUB     #2,(6)          ;FORM PC OF EMT INSTRUCTION
001612                                              MOV     (6),PCADD       ;GET PC OF EMT INSTRUCTION
000000                                              MOV     @(6),(6)        ;GET EMT INSTRUCTION
000001                                              CLRB    1(6)            ;CLEAR MSH OF EMT INSTRUCTION
                                                    ASL     (6)             ;SHIFT EMT IDENTIFIER
001540                                              ADD     #EMTTAB,(6)
                                                    MOV     @(6)+,%7        ;GO TO PROPER EMT
```

```
                                        ;SAVE REGS 0 TO 4 SUBROUTINE.
002574  012637  002630  SAVRG:  MOV     (6)+,SVRPC      ;SAVE PC AND PSW.
002600  012637  002632          MOV     (6)+,SVRPSW
002604  010446                  MOV     %4,-(6)         ;SAVE REGS 0 - 4
002606  010346                  MOV     %3,-(6)         ;IN STACK.
002610  010246                  MOV     %2,-(6)
002612  010146                  MOV     %1,-(6)
002614  010046                  MOV     %0,-(6)
002616  013746  002632          MOV     SVRPSW,-(6)     ;RESTORE PC AND PSW.
002622  013746  002630          MOV     SVRPC,-(6)
002626  000002                  RTI                     ;EXIT.
002630  000000  SVRPC:  OPEN
002632  000000  SVRPSW: OPEN

                                        ;RESTORE REGS 0 TO 4 SUBROUTINE.
002634  012637  002670  RSTRG:  MOV     (6)+,RSTPC      ;SAVE PC AND PSW.
002640  012637  002672          MOV     (6)+,RSTPSW
002644  012600                  MOV     (6)+,%0         ;RESTORE REGS 0 - 4
002646  012601                  MOV     (6)+,%1         ;FROM STACK.
002650  012602                  MOV     (6)+,%2
002652  012603                  MOV     (6)+,%3
002654  012604                  MOV     (6)+,%4

002656  013746  002672          MOV     RSTPSW,-(6)     ;RESTORE PC AND PSW.
002662  013746  002670          MOV     RSTPC,-(6)
002666  000002                  RTI                     ;EXIT
002670  000000  RSTPC:  OPEN
002672  000000  RSTPSW: OPEN

                                        ;ROUTINE TO ISSUE RESET.
002674  012700  052525  SRSETT: MOV     #52525,%0               ;DATA TO R0.
002700  005100                  COM     %0              ;COMPLEMENT (R0).
002702  010037  002676          MOV     %0,SRSETT+2     ;(R0) TO SRSETT+2.
002706  000005                  RESET                   ;ISSUE RESET. (R0) IS
002710  000002                  RTI                     ;DISPLAYED. EXIT.

                                        ;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.
002712  013700  002760  RNGEN:  MOV     RP1,%0
002716  006100                  ROL     %0
002720  006100                  ROL     %0
002722  063700  002762          ADD     RP2,%0
002726  010037  002760          MOV     %0,RP1
002732  006100                  ROL     %0
002734  006100                  ROL     %0
002736  063700  002762          ADD     RP2,%0
002742  006100                  ROL     %0
002744  006100                  ROL     %0
002746  010037  002762          MOV     %0,RP2
002752  013700  002760          MOV     RP1,%0
002756  000207                  RTS     %7              ;EXIT. NUMBER IN R0
002760  001233  RP1:    1233
002762  007622  RP2:    7622
                                        ;SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER.
002764  011600  TYP:    MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS.
002766  062716  000002          ADD     #2,@%6          ;SET UP EXIT.
002772  011000                  MOV     @%0,%0          ;ADDRESS OF MESSAGE TO R0.
```

```
  992 002774 112037 003074          TYPA:  MOVB    (0)+,TYPDAT     ;GET CHARACTER
  993 003000 122737 000100 003074          CMPB    #100,TYPDAT     ;CHECK FOR"@"CHARACTER
  994 003006 001001                        BNE     TYPC            ;BRANCH IF NOT"@".
  995 003010 000002                        RTI                     ;TERMINATOR CHAR. DONE. EXIT.
  996 003012 122737 000045 003074  TYPC:  CMPB    #45,TYPDAT      ;CHECK FOR"%".
  997 003020 001412                        BEQ     TYPF            ;BRANCH IF"%"
  998 003022 004737 003030                 JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
  999 003026 000762                        BR      TYPA
 1000 003030 113777 003074 176544  TYPD:  MOVB    TYPDAT,@TPDBR   ;OUTPUT CHARACTER TO PRINTER
 1001 003036 105777 176536                TSTB    @TPCSR          ;WAIT FOR DONE FLAG.
 1002 003042 100375                       BPL     .-4
 1003 003044 000207                       RTS     %7              ;EXIT
 1004 003046 112737 000015 003074  TYPF:  MOVB    #15,TYPDAT      ;MOVE CARRIAGE RETURN CODE TO TYPDAT
 1005 003054 004737 003030                JSR     %7,TYPD         ;GO TYPE CHAR.
 1006 003060 112737 000012 003074  TYPG:  MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT.
 1007 003066 004737 003030                JSR     %7,TYPD         ;GO TYPE CHAR.
 1008 003072 000740                       BR      TYPA
 1009 003074 000000                TYPDAT: OPEN
 1010
 1011
 1012                                     ;SUBROUTINE TO INITIALIZE STACK POINTER AND SET PROCESSOR PRIORITY
 1013                                     ;LEVEL 7
 1014 003076 012777 001106 176152  INIT:  MOV     #CAT,@BASREG    ;INITIALIZE THE BASE REGISTER
 1015 003104 012737 000340 177776         MOV     #PRTY7,PSW              ;SET PRIORITY LEVEL 7
 1016 003112 011637 001100                MOV     (SP),SPBOT      ;GET RETURN ADDRESS
 1017 003116 012706 001100                MOV     #SPBOT,SP       ;SET BOTTOM OF THE STACK
 1018 003122 000176 000000                JMP     @(SP)           ;RETURN
 1019
 1020
 1021                                     ;SUBROUTINE TO GET DM11 PARAMETERS
 1022
 1023                                     ;VECTOR ADDRESS
 1024 003126 000240                DMPAR:  NOP                     ;BEGIN
 1025 003130 004737 003646                JSR     7,OVRLAY        ;PUT HALT,.+2 IN VECTOR AREA
 1026 003134 104000                       TYPE                    ;ASK USER FOR RECEIVER INT. VECTOR
 1027 003136 017064                       WHERE                   ;OF UNIT UNDER TEST
 1028 003140 004537 003702                JSR     5,RECO          ;GET VECTOR AND PUT IT
 1029 003144 000000                VECTOR: 0                       ;HERE
 1030 003146 005737 003144                TST     VECTOR
 1031 003152 001003                       BNE     VECOK
 1032 003154 012737 000300 003144         MOV     #300,VECTOR     ;SET VECTOR = TO 0300
 1033 003162 023727 003144 000300  VECOK: CMP     VECTOR,#300     ;IS VECTOR HIGHER OR
 1034 003170 103003                       BHIS    VECOKB          ;EQUAL TO 0300
 1035 003172 104000                VECOKA: TYPE                    ;TYPE '?'
 1036 003174 017241                       MI
 1037 003176 000753                       BR      DMPAR           ;ASK FOR ANOTHER VECTOR
 1038 003200 023727 003144 000770  VECOKB: CMP     VECTOR,#770     ;IS VECTOR = TO OR
 1039 003206 101371                       BHI     VECOKA          ;LESS THAN 770
 1040 003210 032737 000007 003144         BIT     #7,VECTOR       ;LSB OF VECTOR MUST BE ALL 0'S
 1041 003216 001365                       BNE     VECOKA
 1042 003220 013737 003144 001260         MOV     VECTOR,CLKINT
 1043 003226 062737 000004 003144         ADD     #4,VECTOR
 1044 003234 013737 003144 001264         MOV     VECTOR,XMTINT
 1045
 1046                                     ;UNIT NUMBER
 1047 003242 104000                DMPARB: TYPE
```

```
1048  003244  012740                        WHICH
1049  003246  004537  003702                JSR     5,RECD          ;GET UNIT AND PUT IT
1050  003252  000000                UNIT:   0                       ;HERE
1051  003254  023727  003252  000017        CMP     UNIT,#17        ;UNIT SELECTED MUST BE
1052  003262  101403                        BLOS    UNTOKA          ;BETWEEN 0 & 17
1053  003264  104000                        TYPE
1054  003266  017241                        M1
1055  003270  000764                        BR      DMPARB
1056  003272  006337  003252        UNTOKA: ASL     UNIT
1057  003276  006337  003252                ASL     UNIT
1058  003302  006337  003252                ASL     UNIT
1059  003306  012702  000004                MOV     #4,%2
1060  003312  012701  001250                MOV     #CSR,%1
1061  003316  042711  000370        UNTOKB: BIC     #370,(1)        ;FORM ADDRESSES OF
1062  003322  063721  003252                ADD     UNIT,(1)+       ;REGISTERS OF UNIT SELECTED
1063  003326  005302                        DEC     %2
1064  003330  001372                        BNE     UNTOKB
1065
1066  003332  012702  000004                MOV     #4,%2
1067  003336  012703  001250                MOV     #CSR,%3
1068  003342  012701  001622                MOV     #.CSR,%1
1069  003346  012311        UNTOKC: MOV     (3)+,(1)        ;FORM ODD BYTE ADDRESSES
1070  003350  005221                        INC     (1)+
1071  003352  005302                        DEC     %2
1072  003354  001374                        BNE     UNTOKC
1073                        ;CHARACTER LENGTH
1074  003356  104000        DMPARC: TYPE
1075  003360  017155                        LEVEL
1076  003362  004537  003702                JSR     5,RECD          ;GET LENGTH AND PUT IT
1077  003366  000000        LENGTH: 0                       ;HERE
1078  003370  005737  003366                TST     LENGTH
1079  003374  001003                        BNE     LENOKA
1080  003376  012737  000010  003366        MOV     #8.,LENGTH
1081  003404  023727  003366  000005 LENOKA: CMP     LENGTH,#5       ;CHARACTER LENGTH SELECTED MUST
1082  003412  103003                        BHIS    LENOKC          ;BE BETWEEN 5-8
1083  003414  104000        LENOKB: TYPE                    ;CARRIAGE RETURN SELECTS 8
1084  003416  017241                        M1
1085  003420  000756                        BR      DMPARC
1086  003422  023727  003366  000010 LENOKC: CMP     LENGTH,#8.
1087  003430  101371                        BHI     LENOKB
1088  003432  162737  000005  003366        SUB     #5,LENGTH
1089  003440  006337  003366                ASL     LENGTH
1090  003444  013701  003366                MOV     LENGTH,%1
1091  003450  016137  003462  001304        MOV     LENOKD(1),@#CARMSK      ;SET CHARACTER LENGTH MASK
1092  003456  000240                        NOP
1093  003460  000207                        RTS     7               ;EXIT PARAMETERS ROUTINE
1094
1095                        ;THE BELOW TABLE REPRESENTS THE CHARACTER LENGTH MASK FOR 5,6,7, AND 8
1096                        ;BITS PER CHARACTER RESPECTIVELY.
1097  003462  177740        LENOKD: 177740
1098  003464  177700                177700
1099  003466  177600                177600
1100  003470  177400                177400
1101
1102                        ;CALCULATE MACHINE TIME TO TRANSMIT ONE CHARACTER
1103  003472  005037  003642 TIMER:  CLR     TIME1
```

```
1104  003476  012737  177777  001146          MOV     #-1,WCT         ;SET UP TO TRANSMIT
1105  003504  012737  016350  001106          MOV     #OUTBUF,CAT     ;1 CHARACTER ON LINE 1
1106  003512  012777  003612  175540          MOV     #TIMEC,@CLKINT  ;LOAD RECEIVER INTERRUPT
1107  003520  012777  000340  175534          MOV     #PRTY7,@CLKLVL  ;AND PRIORITY
1108  003526  012777  000001  175516          MOV     #LBIT0,@BAR     ;START TRANSMITTING
1109  003534  012777  000105  175506          MOV     #BIT6+BIT2+BIT0,@CSR    ;SET IE BIT
1110  003542  005037  177776                  CLR     @#PSW           ;SET PROCESSER PRIORITY LEVEL = 0
1111  003546  012737  000044  001604  TIMEA:  MOV     #44,COUNT
1112  003554  062737  000001  003642          ADD     #1,TIME1        ;INCREMENT MACH. TIME TO TRANSMIT
1113  003562  001007                          BNE     TIMEB
1114  003564  005077  175460                  CLR     @CSR
1115  003570  012737  000340  177776          MOV     #PRTY7,PSW      ;SET PROCESSER PRIORITY LEVEL = 7
1116  003576  104001                          ERROR                   ;TRANSMITTER FAILED TO INTERRUPT
1117  003600  000734                          BR      TIMER
1118  003602  005337  001604          TIMEB:  DEC     COUNT
1119  003606  001375                          BNE     .-4
1120  003610  000756                          BR      TIMEA
1121  003612  005077  175432          TIMEC:  CLR     @CSR
1122  003616  013737  003642  003644          MOV     TIME1,TIME14
1123  003624  006037  003644                  ROR     TIME14
1124  003630  000241                          CLC
1125  003632  006037  003644                  ROR     TIME14
1126  003636  022626                          POPSP2                  ;RESTORE STACK POINTER
1127  003640  000207                          RTS     7               ;EXIT TIME CALCULATION ROUTINE
1128
1129  003642  000000          TIME1:  OPEN                            ;CONTAINS MACHINE TIME TO XMIT 1 CHAR
1130  003644  000000          TIME14: OPEN                            ;CONTAIN TIME TO XMIT 2-4 CHAR
1131
1132                          ;SUBROUTINE TO PUT HALT,.+2 IN VECTOR AREA (0300-1000)
1133  003646  012701  000300  OVRLAY: MOV     #300,%1
1134  003652  012702  000302          MOV     #302,%2
1135  003656  010221          1$:     MOV     %2,(1)+
1136  003660  005021                  CLR     (1)+
1137  003662  020227  000776                  CMP     %2,#776
1138  003666  001403                          BEQ     2$
1139  003670  062702  000004                  ADD     #4,%2
1140  003674  000770                          BR      1$
1141  003676  000240          2$:     NOP
1142  003700  000207                  RTS     7               ;EXIT
1143
1144
1145                          ;SUBROUTINE TO RECEIVE DATA
1146                          ;THIS SUBROUTINE RECEIVES DATA FROM THE KEYBOARD (UP TO SIX OCTAL
1147                          ;DIGITS AND PLACES THEM INTO THE ADDRESS FOLLOWING THE SUBROUTINE
1148                          ;CALL (JSR 5,RECD).  NO REGISTER CONTENTS ARE DISTURBED.
1149
1150
1151                          ;SUBROUTINE TO INPUT DATA FROM TTY
1152
1153  003702  010046          RECD:   MOV     R0,-(SP)
1154  003704  005015          1$:     CLR     (5)                     ;CLEAR OLD DATA
1155  003706  012737  000007  004060          MOV     #7,CNT          ;SET CHAR COUNT
1156  003714  105777  175654          2$:     TSTB    @TKCSR          ;WAIT FOR CHAR
1157  003720  100375                          BPL     2$
1158  003722  117700  175650                  MOVB    @TKDBR,R0
1159  003726  142700  000200                  BICB    #200,R0         ;STRIP OFF PARITY
```

```
1160  003732  110077  175644            MOVB    R0,@TPDBR       ;ECHO CHARACTER
1161  003736  122700  000025            CMPB    #25,R0          ;IS IT A ↑U
1162  003742  001443                    BEQ     5$              ;BRANCH IF YES
1163  003744  122700  000015            CMPB    #15,R0          ;IS IT A <CR>
1164  003750  001415                    BEQ     6$              ;BRANCH IF YESS
1165  003752  142700  000060            BICB    #60,R0
1166  003756  132700  000110            BITB    #110,R0         ;CHECK FOR 0-7 (8)
1167  003762  001031                    BNE     7$              ;BRANCH IF NOT
1168  003764  006315                    ASL     (5)
1169  003766  006315                    ASL     (5)
1170  003770  006315                    ASL     (5)             ;SHIFT DATA
1171  003772  150015                    BISB    R0,(5)          ;INSET NEW CHAR
1172  003774  005337  004060            DEC     CNT
1173  004000  001422                    BEQ     7$              ;ONLY 6 CHAR'S PLEASE
1174  004002  000744                    BR      2$              ;NEXT CHARACTER
1175  004004  105777  175570    6$:     TSTB    @TPCSR
1176  004010  100375                    BPL     6$              ;WAIT FOR READY
1177  004012  012777  000012  175562    MOV     #12,@TPDBR      ;TYPE <LF>
1178  004020  105777  175554    8$:     TSTB    @TPCSR
1179  004024  100375                    BPL     1$              ;WAIT FOR READY
1180  004026  005077  175550            CLR     @TPDBR          ;LOAD CHAR
1181  004032  105777  175542    9$:     TSTB    @TPCSR
1182  004036  100375                    BPL     9$
1183  004040  005725                    TST     (R5)+           ;ADJUST R5
1184  004042  012600                    MOV     (SP)+,R0        ;RESTORE R0
1185  004044  000205                    RTS     R5
1186  004046  104000            7$:     TYPE
1187  004050  017241                    M1
1188  004052  104000            5$:     TYPE
1189  004054  017135                    $CTLU
1190  004056  000712                    BR      1$                      ;START OVER
1191  004060  000000            CNT:    0
1192
1193                            ;POWER FAIL ROUTINE
1194  004062  012737  004072  000024  PFAIL:  MOV     #PWRUP,24
1195  004070  000000                    HALT
1196
1197                            ;POWER UP SUBROUTINE
1198  004072  000005            PWRUP:  RESET                   ;GIVE TELEPRINTER TIME TO START
1199  004074  012706  001100            MOV     #SPBOT,%6
1200  004100  104001                    ERROR                   ;TYPE POWER FAIL ERROR
1201  004102  000137  002206            JMP     @#RSTAT2        ;GO RESTART PROGRAM
1202
1203                            ;LINE TEST SUBROUTINE: THIS LINE TEST PROVIDES SEVERAL TESTS ON A DM11 LINE.
1204                            ;THE SUBROUTINE IS CALLED BY JSR 5, LNTST.  THIS INSTRUCTION PROVIDES THE
1205                            ;LINE BIT AND LINE NUMBER.  THE FOLLOWING LINE TESTS ARE PERFORMED:
1206                            ;WAITS UNTIL CHARACTER SHOULD HAVE BEEN TRANSMITTED, THEN TESTS
1207
1208                                    ;THAT BAR BIT CLEARED           ;DO NEXT TEST IF ERROR
1209                                    ;READY SET                      ;DO NEXT TEST IF ERROR
1210                                    ;WORD COUNT WENT TO 0           ;DO NEXT TEST IF ERROR
1211                                    ;CURRENT ADDRESS DID NOT INCREMENT ;DO NEXT TEST IF ERROR
1212                                    ;INTERRUPTS TO CORRECT VECTOR   ;DO NEXT TEST IF ERROR (NO INTERRUPT)
1213                                    ;READY BIT CAN BE CLEARED       ;END OF TEST
1214  004106  012537  016176    XMTTST: MOV     (5)+,@#LINE     ;GET LINE NUMBER
1215  004112  004737  006330            JSR     7,GTLINB        ;GO FORM LINE BIT (FOR BAR)
```

```
1216  004116  005037  001302          CLR     XMTDAT
1217  004122  004537  005532    1$:   JSR     5,@#XMITD       ;GO TO TRANSMIT SUBROUTINE
1218  004126  177777                  -1                      ;TRANSMIT ONE CHARACTER
1219  004130  012703  000010          MOV     #10,%3          ;WAIT IN
1220  004134  005002                  CLR     %2              ;THIS
1221  004136  005302          2$:     DEC     %2              ;LOOP
1222  004140  001376                  BNE     .-2             ;UNTIL THE
1223  004142  005303                  DEC     %3              ;TRANSMITTER
1224  004144  001374                  BNE     2$              ;IS FINISHED
1225  004146  017737  175100  001300  MOV     @BAR,RCVDAT     ;BAR SHOULD NOW BE CLEAR
1226  004154  001401                  BEQ     3$              ;BRANCH IF IT IS
1227  004156  104011                  ERROR1                  ;ERROR! BAR BIT FAILED TO CLEAR
1228  004160  005777  175064    3$:   TST     @CSR            ;TEST READY BIT SHOULD BE SET
1229  004164  100401                  BMI     4$              ;BRANCH IF SET
1230  004166  104001                  ERROR                   ;ERROR! READY NOT SET
1231  004170  013701  016176    4$:   MOV     @#LINE,%1       ;GET LINE NUMBER
1232  004174  016137  001146  001300  MOV     WCT(1),RCVDAT   ;WORD COUNT SHOULD BE 0
1233  004202  001401                  BEQ     5$
1234  004204  104011                  ERROR1                  ;ERROR! WORD COUNT NOT EQUAL TO 0
1235  004206  012737  016350  001302  5$:   MOV     #OUTBUF,XMTDAT  ;
1236  004214  016137  001106  001300  MOV     CAT(1),RCVDAT   ;CURRENT ADDRESS SHOULD NOT HAVE INCREMENTED
1237  004222  023737  001300  001302  CMP     RCVDAT,XMTDAT   ;
1238  004230  001401                  BEQ     6$
1239  004232  104011                  ERROR1                  ;ERROR! CURRENT ADDR. DID NOT INCREMENT
1240  004234  012777  004266  175022  6$:   MOV     #7$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
1241  004242  052777  010000  175000        BIS     #BIT12,@CSR     ;ENABLE TRANSMITTER INTERRUPT
1242  004250  005037  177776          CLR     @#PSW           ;SET PROCESSOR PRIORITY =0
1243  004254  000240                  NOP
1244  004256  012737  000340  177776  MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1245  004264  104001                  ERROR                   ;TRANSMITTER FAILED TO INTERRUPT OR
1246                          ;INTERRUPTED TO WRONG LOCATION AND HALTED WITH ADDRESS +2 DISPLAYED.
1247  004266  022626          7$:     CMP     (6)+,(6)+       ;RESET STACK PTR
1248  004270  012737  000340  177776  MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1249  004276  042777  110000  174744  BIC     #BIT12!BIT15,@CSR       ;CLEAR XMIT IE & READY BITS
1250  004304  005777  174740          TST     @CSR            ;TEST THAT READY CLEARED
1251  004310  100001                  BPL     8$              ;GO TO EXIT
1252  004312  104001                  ERROR                   ;ERROR! READY FAILED TO CLEAR
1253  004314  005726          8$:     TST     (6)+            ;RESET STACK PTR
1254  004316  104006                  SCOPE                   ;SCOPE
```

```
1255
1256
1257                                    ;RECEIVER LINE TESTS
1258                                    ;THE RECEIVER LINE TEST SUBROUTINE IS ENTERED WITH
1259                                    ;A JSR 5, RCVTST INSTRUCTION FOLLOWED BY THE
1260                                    ;LINE BIT AND LINE NUMBER OF THE LINE TO BE
1261                                    ;TESTED.  THE SUBROUTINE PERFORMS THE FOLLOWING
1262                                    ;TEST AS SHOWN BELOW.  IN THE EVENT OF AN ERROR
1263                                    ;THE REMAINING TESTS ARE ABORTED
1264                                    ;TEST SEQUENCE AND ADDRESS TAG
1265                                    ;
1266                                    ;       CHARACTER DONE SETS                             RTSTA
1267                                    ;       CHARACTER DONE CAUSES INTERRUPT                 RTSTB
1268                                    ;       CHARACTER DONE CAN BE CLEARED                   RTSTC
1269                                    ;       TUMBLE TABLE ENTRY IS CORRECT                   RTSTD
1270                                    ;       NO ENTRY IN NEXT TABLE ADDRESS                  RTSTE
1271                                    ;       HARDWARE TABLE POINTER INCREMENTED              RTSTF
1272                                    ;       NEXT ENTRY WAS CORRECT                          RTSTG
1273
1274                                    ;NOTES: IF THE HARDWARE PROVIDES AN INCORRECT VECTOR
1275                                    ;       ADDRESS THE PROGRAM WILL HALT AND DISPLAY
1276                                    ;       THE INCORRECT VECTOR+2 IN THE ADDRESS LIGHTS.
1277
1278  004320  012737  177777  016350  RCVTST: MOV     #-1,OUTBUF      ;LOAD ALL 1'S INTO OUTPUT BUFFER
1279  004326  005037  001306          CLR     TUMTAB          ;CLEAR THE FIRST
1280  004332  005037  001310          CLR     TUMTAB+2        ;TWO TUMBLE TABLE ADDRESSES
1281  004336  012737  000340  177776  MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1282  004344  012537  016176          MOV     (5)+,LINE       ;GET LINE NUMBER
1283  004350  004537  005532          JSR     5,@#XMITD       ;TRANSMIT 1 CHARACTER (0'S)
1284  004354  177777                  -1                      ;ON LINE SPECIFIED BY JSR
1285  004356  052777  000001  174664  BIS     #BIT0,@CSR      ;SET GO BIT
1286  004364  005777  174660          TST     @CSR            ;WAIT FOR TRANSMITTER
1287  004370  100375                  BPL     .-4             ;TO TRANSMIT 1 CHAR.
1288  004372  042777  100000  174650  BIC     #BIT15,@CSR     ;CLEAR TRANSMITTER READY FLAG
1289  004400  005046                  CLR     -(SP)           ;SET WATCH DOG TIMER
1290  004402  105777  174642  1$:     TSTB    @CSR            ;TEST CHAR. DONE FLAG
1291  004406  100404                  BMI     2$              ;BRANCH IF SET
1292  004410  005216                  INC     (SP)            ;WAIT FOR THE FLAG
1293  004412  001373                  BNE     1$
1294  004414  104001                  ERROR                   ;ERROR! CHAR. DONE FLAG FAILED TO SET
1295  004416  000550                  BR      8$              ;GO TO EXIT
1296  004420  005726          2$:     TST     (SP)+           ;RESTORE STACK PTR
1297  004422  012777  004456  174630  MOV     #3$,@CLKINT     ;LOAD RECEIVER INTERRUPT VEC. ADRS.
1298  004430  052777  000100  174612  BIS     #BIT6,@CSR      ;SET RECEIVER IE BIT
1299  004436  005037  177776          CLR     @#PSW           ;ENABLE INTERRUPTS
1300  004442  000240                  NOP
1301  004444  012737  000340  177776  MOV     #PRTY7,PSW      ;LOCK OUT INTERRUPTS
1302  004452  104001                  ERROR                   ;RECEIVER FAILED TO INTERRUPT
1303  004454  000531                  BR      8$              ;GO TO EXIT
1304  004456  012737  000340  177776  3$:     MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1305  004464  022626                  CMP     (6)+,(6)+
1306  004466  042777  000300  174554  BIC     #BIT7+BIT6,@CSR ;CLEAR CHAR. DONE FLAG
1307  004474  105777  174550          TSTB    @CSR            ;TEST THAT CHAR DONE FLAG CLEARED
1308  004500  100002                  BPL     4$              ;BRANCH IF CHAR. DONE FLAG CLEARED
1309  004502  104001                  ERROR                   ;ERROR! CHAR. DONE FAILED TO CLEAR
1310  004504  000515                  BR      8$              ;GO TO EXIT
```

```
1311  004506  013737  001306  001300  4$:     MOV     TUMTAB,RCVDAT   ;GET TUMBLE TABLE ENTRY
1312  004514  042737  020000  001300          BIC     #BIT13,RCVDAT   ;CLEAR PARITY INDICATOR
1313  004522  012737  000377  001302          MOV     #377,XMTDAT     ;LOAD XMTDAT WITH  TRANSMITTED DATA
1314  004530  043737  001304  001302          BIC     CARMSK,XMTDAT   ;CLEAR NON TRANSMITTED BITS
1315  004536  153737  016176  001303          BISB    LINE,XMTDAT+1   ;LOAD XMTDAT WITH PROPER LINE #
1316  004544  052737  100000  001302          BIS     #BIT15,XMTDAT   ;SET VALID DATA ENTRY BIT IN XMTDAT
1317  004552  023737  001300  001302          CMP     RCVDAT,XMTDAT   ;COMPARE TUMBLE TABLE ENTRY (RCVDAT) &
1318                                                                  ;CORRECT RESULT (XMTDAT)
1319
1320  004560  001402                          BEQ     5$
1321  004562  104011                          ERROR!                  ;ERROR! INCORRECT TUMBLE TABLE
1322  004564  000465                          BR      8$              ;ENTRY; GO TO EXIT
1323  004566  005037  001306          5$:     CLR     TUMTAB
1324  004572  013737  001310  001300          MOV     TUMTAB+2,RCVDAT ;GET NEXT ENTRY
1325  004600  001404                          BEQ     6$              ;BRANCH IF ALL 0'S
1326  004602  005037  001302                  CLR     XMTDAT
1327  004606  104011                          ERROR!                  ;ERROR! FALSE ENTRY IN NEXT
1328  004610  000453                          BR      8$              ;TUMBLE TABLE ADDRESS
1329  004612  004537  005532          6$:     JSR     5,@#XMITD       ;TRANSMIT 1 CHARACTER (ALL 1'S)
1330  004616  177777                          -1                      ;ON LINE SPECIFIED BY JSR
1331  004620  005777  174424                  TST     @CSR            ;WAIT FOR TRANSMITTER
1332  004624  100375                          BPL     .-4             ;READY FLAG
1333  004626  105777  174416                  TSTB    @CSR            ;TEST FOR THE DONE FLAG
1334  004632  100375                          BPL     .-4
1335  004634  042777  000200  174406          BIC     #BIT7,@CSR      ;CLEAR CHAR. DONE FLAG
1336  004642  013737  001306  001300          MOV     TUMTAB,RCVDAT   ;TEST THAT HARDWARE TUMBLE
1337  004650  001404                          BEQ     7$              ;TABLE POINTER INCREMENTED (+2)
1338  004652  005037  001302                  CLR     XMTDAT
1339  004656  104011                          ERROR!                  ;ERROR! TUMBLE TABLE POINTER DID
1340  004660  000427                          BR      8$              ;NOT INCREMENT; GO TO EXIT
1341  004662  013737  001310  001300  7$:     MOV     TUMTAB+2,RCVDAT ;GET TUMBLE TABLE ENTRY
1342  004670  042737  020000  001300          BIC     #BIT13,RCVDAT   ;CLEAR PARITY INDICATOR
1343  004676  012737  000377  001302          MOV     #377,XMTDAT     ;LOAD XMTDAT WITH TRANSMITTED DATA
1344  004704  043737  001304  001302          BIC     CARMSK,XMTDAT   ;CLEAR NON-TRANSMITTED BITS
1345  004712  153737  016176  001303          BISB    LINE,XMTDAT+1   ;LOAD LINE # INTO XMTDAT
1346  004720  052737  100000  001302          BIS     #BIT15,XMTDAT   ;SET VALID DATA ENTRY BIT INTO XMTDAT
1347  004726  023737  001300  001302          CMP     RCVDAT,XMTDAT   ;COMPARE TUMBLE TABLE ENTRY (RCVDAT) &
1348                                                                  ;CORRECT RESULT (XMTDAT)
1349  004734  001401                          BEQ     8$
1350  004736  104011                          ERROR!                  ;ERROR! 2ND TUMBLE TABLE ENTRY
1351  004740  104006                  8$:     SCOPE                   ;WAS INCORRECT; SCOPE
1352
1353
1354                                  ;SUBROUTINE TO TEST BREAK OPERATION
1355                                  ;THE TRANSMITTER WILL TRANSMIT THE BREAK FOR TWO CHARACTER
1356                                  ;TIMES AND THEN THE FOLLOWING TESTS WILL BE PERFORMED
1357                                  ;       A VALID DATA ENTRY WAS MADE     BKTSTB
1358                                  ;       BREAK BIT SET                   BKTSTC
1359                                  ;       DATA WAS ALL 0'S                BKTSTD
1360  004742  012777  000001  174300  BRKTST: MOV     #1,@CSR         ;SET THE GO BIT
1361  004750  011577  174300                  MOV     (5),@BKCSR      ;SET THE BREAK BIT
1362  004754  105777  174270                  TSTB    @CSR            ;WAIT FOR THE RECEIVER TO
1363  004760  100375                          BPL     .-4             ;RECEIVE BREAK
1364  004762  042777  000200  174260          BIC     #BIT7,@CSR      ;CLEAR FLAG
1365  004770  105777  174254                  TSTB    @CSR            ;WAIT FOR THE RECEIVER TO
1366  004774  100375                          BPL     .-4             ;TO RECEIVE BREAK
```

```
1367  004776  042777  000200  174244          BIC     #BIT7,@CSR      ;CLEAR FLAG
1368  005004  005077  174244                  CLR     @BKCSR          ;CLEAR BREAK BIT
1369  005010  005737  001306          1$:     TST     TUMTAB          ;TEST FOR VALID DATA ENTRY
1370  005014  100402                          BMI     2$
1371  005016  104001                          ERROR                   ;ERROR! NO VALID DATA ENTRY
1372  005020  000421                          BR      4$              ;GO TO EXIT
1373  005022  032737  040000  001306  2$:     BIT     #BIT14,TUMTAB   ;TEST THAT BREAK BIT IS SET
1374  005030  001002                          BNE     3$              ;IN TUMBLE TABLE
1375  005032  104001                          ERROR                   ;ERROR! BREAK BIT FAILED TO SET
1376  005034  000413                          BR      4$              ;GO TO EXIT
1377  005036  105737  001306          3$:     TSTB    TUMTAB          ;TEST THAT DATA IS ALL 0'S
1378  005042  001410                          BEQ     4$
1379  005044  005037  001300                  CLR     RCVDAT
1380  005050  113737  001306  001300          MOVB    TUMTAB,RCVDAT   ;GET RECEIVED DATA
1381  005056  005037  001302                  CLR     XMTDAT
1382  005062  104011                          ERROR1                  ;ERROR! DATA WAS NOT ALL 0'S
1383  005064  104006                  4$:     SCOPE                   ;SCOPE
1384
1385
1386                                  ;SUBROUTINE TO TRANSMIT & RECEIVE ON ALL LINES THE DELAY BETWEEN
1387                                  ;TRANSMITTING ON A LINE IS SUPPLIED BY THE CALLING JSR INSTRUCTION.
1388                                  ;NOTE NO DATA CHECKING IS PERFORMED BY THIS TEST
1389  005066  012537  001604          DLYXMT: MOV     (5)+,@#COUNT    ;GET CHARACTER DELAY COUNT
1390  005072  005037  001302                  CLR     XMTDAT
1391  005076  004537  005510                  JSR     5,BMOVE         ;LOAD OUTPUT BUFFER WITH DATA
1392  005102  017632                          MSG1                    ;TO BE TRANSMITTED
1393  005104  016350                          OUTBUF
1394  005106  000100                          64.
1395  005110  012737  000001  001272          MOV     #LBIT0,@#LINBIT
1396  005116  005037  016176                  CLR     @#LINE
1397  005122  012777  000001  174120  1$:     MOV     #BIT0,@CSR      ;SET THE GO BIT
1398  005130  004537  005532          2$:     JSR     5,@#XMTBYT      ;TRANSMIT 64. CHAR.
1399  005134  177700                          -64.                    ;ON A LINE
1400  005136  013737  003642  005152          MOV     @#TIME1,4$
1401  005144  013704  001604                  MOV     @#COUNT,%4      ;GET CHARACTER DELAY COUNT
1402  005150  104400                  3$:     DELAY
1403  005152  000000                  4$:     0
1404  005154  005304                          DEC     %4
1405  005156  001374                          BNE     3$
1406  005160  062737  000002  016176          ADD     #2,LINE         ;FORM NEXT LINE NUMBER
1407  005166  006337  001272                  ASL     LINBIT          ;SHIFT LINE BIT
1408  005172  103356                          BCC     2$              ;BRANCH IF ALL LINES NOT DONE
1409  005174  012704  000100                  MOV     #64.,%4
1410  005200  013737  003642  005210          MOV     TIME1,6$
1411  005206  104400                  5$:     DELAY
1412  005210  000000                  6$:     0
1413  005212  005304                          DEC     %4
1414  005214  001374                          BNE     5$
1415  005216  017737  174030  001300          MOV     @BAR,RCVDAT     ;GET & TEST BAR DATA
1416  005224  001402                          BEQ     7$              ;EXIT IF DONE
1417  005226  104011                          ERROR1                  ;ERROR! BAR SHOULD'VE BEEN CLEAR
1418  005230  000413                          BR      8$
1419  005232  022777  100201  174010  7$:     CMP     #100201,@CSR    ;TEST THAT ONLY DONE,GO,& READY BITS ARE SET
1420  005240  001407                          BEQ     8$
1421  005242  012737  100201  001302          MOV     #100201,XMTDAT
1422  005250  017737  173774  001300          MOV     @CSR,RCVDAT     ;GET CSR CONTENTS
```

```
                                        ERROR1                  ;INCORRECT CSR COMTENTS
005260  005726                  8$:   POPSP                   ;RESET THE STACK
005262  104006                        SCOPE                   ;SCOPE

                              ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
005264  011637  005334        DLY:  MOV     (6),3$            ;GET DELAY COUNT ADDRESS.
005270  062716  000002              ADD     #2,(6)            ;SET UP EXIT ADDRESS
005274  017737  000034  005334      MOV     @3$,3$            ;GET DELAY COUNT
005302  001413                      BEQ     2$                ;EXIT IF NO DELAY
005304  012737  000050  005336  1$:  MOV     #50,4$
005312  162737  000001  005334      SUB     #1,3$
005320  001404                      BEQ     2$
005322  005337  005336              DEC     4$
005326  001375                      BNE     .-4
005330  000765                      BR      1$
005332  000002              2$:   RTI                       ;EXIT
005334  000000              3$:   OPEN                      ;CONTAINS DELAY COUNT
005336  000000              4$:   OPEN                      ;CONTAINS DELAY ROUTINE CONSTANT

                              ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
005340  012737  177777  005362  INBIN:  MOV     #-1,RIND        ;SET ALL VARIABLES
005346  004537  005510              JSR     %5,BMOVE        ;TO MINUS 1.
                                        RIND
                                        RIND+1
                                        5
005360  000207                      RTS     %7              ;EXIT
005362  000000              RIND:   OPEN
005364  000000              PTO:    OPEN
005366  000000              PT1:    OPEN

                              ;SPECIAL BINARY COUNT PATTEFN SUBROUTINE. EXITS WITH BIN CHAR IN R1
005370  013737  005364  005366  GTBIN:  MOV     PTO,PT1         ;PREVIOUS BIN CHAR TO PT1
005376  005137  005366              COM     PT1
005402  005137  005362              COM     RIND
005406  001002                      BNE     .+6
005410  005237  005366              INC     PT1
005414  042737  177400  005366      BIC     #177400,PT1     ;MASK TO 8 BITS
005422  013737  005366  005364      MOV     PT1,PTO         ;SAVE BIN CHAR IN PTO
005430  013701  005366              MOV     PT1,%1          ;BIN CHAR TO R1.
005434  000240                      NOP
005436  000207                      RTS     %7              ;EXIT.

                              ;OCTAL TO ASCII CONVERT ROUTINE
005440  104007              OACNV:  SAVREG                  ;SAVE REGISTERS ON THE STACK
005442  013504                      MOV     @(5)+,%4        ;GET OCTAL VALUE.
005444  012501                      MOV     (5)+,%1         ;GET DESTINATION ADDR.
005446  012502                      MOV     (5)+,%2         ;GET CONVERT COUNT.
005450  060201                      ADD     %2,%1           ;DEVELOP ADDR TO STORE 1ST CHAR.
005452  010403              OACNVA: MOV     %4,%3
005454  042703  177770              BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT DIGIT.
005460  062703  000060              ADD     #'0,%3          ;CONVERT DIGIT TO ASCII.
005464  110341                      MOVB    %3,-(1)         ;STORE ASCII CHARACTER.
005466  042704  000007              BIC     #7,%4
005472  006004                      ROR     %4
005474  006004                      ROR     %4
005476  006004                      ROR     %4
```

```
1479   005500  005302                          DEC     %2              ;DONE ALL DIGITS?
1480   005502  001363                          BNE     OACNVA          ;BRANCH IF NOT DONE.
1481   005504  104010                          RSTREG                  ;RESTORE THE REGISTERS
1482   005506  000205                          RTS     %5              ;DONE. EXIT.
1483
1484                                   ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1485   005510  104007                  BMOVE:  SAVREG                  ;SAVE REGS.
1486   005512  012501                          MOV     (5)+,%1         ;GET "FROM" ADDRESS
1487   005514  012502                          MOV     (5)+,%2         ;GET "TO" ADDRESS
1488   005516  012503                          MOV     (5)+,%3         ;GET COUNT
1489   005520  112122                  1$:     MOVB    (1)+,(2)+       ;MOVE BYTE
1490   005522  005303                          DEC     %3              ;DECREMENT COUNT
1491   005524  001375                          BNE     1$              ;BRANCH IF NOT DONE.
1492   005526  104010                          RSTREG                  ;RESTORE REGS.
1493   005530  000205                          RTS     %5              ;DONE EXIT
1494
1495                                   ;SUBROUTINE TO TRANSMIT DATA.  SUBROUTINE CALLED BY
1496                                   ;JSR 5,XMITD
1497   005532  010046                  XMITD:  MOV     %0,-(SP)        ;SAVE R0 ON THE STACK
1498   005534  013700  016176                  MOV     @#LINE,%0       ;GET LINE
1499   005540  004737  006330                  JSR     7,@#GTLINB      ;FORM LINE BIT (FOR BAR)
1500   005544  012777  001106  173504          MOV     #CAT,@BASREG    ;INITIALIZE BASE REGISTER
1501   005552  012760  016350  001106          MOV     #OUTBUF,CAT(0)  ;LOAD FIRST CHAR ADDRESS IN CAT
1502   005560  012560  001146                  MOV     (5)+,WCT(0)     ;GET WORD COUNT
1503   005564  053777  001272  173460          BIS     @#LINBIT,@BAR   ;LOAD LINE POSITION INTO BAR
1504   005572  012600                          MOV     (SP)+,%0        ;RESTORE R0
1505   005574  000205                          RTS     5               ;EXIT
1506
1507                                   ;ROUTINE TO TEST A LINE.
1508                                   ;THE LINE TO BE TESTED IS PROVIDED BY THE JSR CALL TO THE ROUTINE.
1509                                   ;100. CHARACTERS ARE TRANSMITTED,RECEIVED AND CHECKED BY THIS ROUTINE.
1510   005576  012537  016176          DATTST: MOV     (5)+,@#LINE     ;GET LINE NUMBER
1511   005602  012737  000144  001604  DATIAA: MOV     #100.,COUNT     ;GET CHARACTER COUNT
1512   005610  012702  016350          MOV     #OUTBUF,%2      ;GET ADDRESS OF OUTPUT BUFFER
1513   005614  004737  005370          1$:     JSR     7,@#GTBIN       ;GET DATA
1514   005620  110122                          MOVB    %1,(2)+         ;LOAD OUTPUT BUFFER WITH DATA
1515   005622  005337  001604                  DEC     COUNT           ;GOT ALL DATA?
1516   005626  001372                          BNE     1$
1517   005630  012701  001306                  MOV     #TUMTAB,%1      ;LOAD TUMBLE TABLE POINTER
1518   005634  005037  001306                  CLR     TUMTAB
1519   005640  004537  005510                  JSR     5,BMOVE         ;CLEAR
1520   005644  001306                          TUMTAB                  ;TUMBLE
1521   005646  001307                          TUMTAB+1                ;TABLE
1522   005650  000177                          177                     ;64 WORDS
1523   005652  012702  016514                  MOV     #INBUF,%2       ;SETUP INPUT BUFFER POINTER
1524   005656  052777  000001  173364          BIS     #BIT0,@CSR      ;SET THE GO BIT
1525   005664  004537  005532                  JSR     5,@#XMITD       ;TRANSMIT
1526   005670  177634                          -100.                   ;100. CHARACTERS
1527   005672  032777  160000  173350  2$:     BIT     #BIT15+BIT14+BIT13,@CSR;TEST IF READY OR ANY ERROR
1528   005700  001004                          BNE     3$              ;FLAGS ARE SET
1529   005702  105777  173342                  TSTB    @CSR            ;WAIT FOR THE RECEIVER
1530   005706  100371                          BPL     2$              ;TO RECEIVE A CHARACTER
1531   005710  000415                          BR      5$
1532   005712  032777  060000  173330  3$:     BIT     #BIT14+BIT13,@CSR;TEST FOR ERROR FLAGS
1533   005720  001403                          BEQ     4$              ;BRANCH NO ERROR
1534   005722  104001                          ERROR
```

```
1535  005724  000137  006244                    JMP     15$             ;GO EXIT
1536  005730  042777  100000  173312  4$:       BIC     #BIT15,@CSR     ;CLEAR TRANSMITTER READY FLAG
1537  005736  105777  173306                    TSTB    @CSR            ;TEST FOR CHARACTER READY
1538  005742  100375                            BPL     .-4
1539  005744  042777  000200  173276  5$:       BIC     #BIT7,@CSR      ;CLEAR CHAR DONE BIT
1540  005752  005711                            TST     (1)
1541  005754  100401                            BMI     .+4             ;TEST FOR VALID ENTRY
1542  005756  104001                            ERROR                   ;REPORT INVALID ENTRY
1543  005760  111122                            MOVB    (1),(2)+        ;MOVE CHAR FROM TUM. TAB. TO INPUT BUFFER
1544
1545                                    ;ROUTINE TO STORE RECEIVED PARITY BIT IN PARITY BIT BUFFER
1546  005762  012705  000001                    MOV     #1,%5           ;GET ROTATE COUNT
1547  005766  000261                            SEC                     ;SET THE CARRY BIT
1548  005770  032711  020000                    BIT     #BIT13,(1)      ;TEST RECEIVED PARITY BIT
1549  005774  001001                            BNE     6$              ;BRANCH IF RECEIVED PARITY WAS ODD
1550  005776  000241                            CLC                     ;CLEAR CARRY BIT
1551  006000  004537  006250          6$:       JSR     5,RORPARBUF     ;ROTATE RECEIVED PARITY INTO PARITY BUFFER
1552
1553                                    ;ROUTINE TO TEST THAT ENTRY IS FOR THE CORRECT LINE.
1554  006004  011137  001270                    MOV     (1),@#TTDAT     ;GET TABLE ENTRY
1555  006010  042737  160777  001270            BIC     #160777,TTDAT   ;CLEAR ALL BUT LINE NUMBER
1556  006016  123737  016176  001271            CMPB    LINE,TTDAT+1    ;COMPARE LINE NUMBERS
1557  006024  001410                            BEQ     7$
1558  006026  013737  016176  001302            MOV     LINE,XMTDAT     ;GET CORRECT LINE # (X2)
1559  006034  013737  001270  001300            MOV     TTDAT,RCVDAT    ;GET LINE # (X2) THAT FALSE DATA CAME IN ON
1560  006042  104011                            ERROR1                  ;ERROR! DATA CAME IN ON A LINE THAT
1561                                                                    ;PROGRAM WAS NOT TRANSMITTING ON.
1562  006044  000477                            BR      15$             ;EXIT TEST
1563  006046  020127  001504          7$:       CMP     %1,#TUMTAB+176  ;IS POINTER AT THE END
1564  006052  001002                            BNE     8$              ;OF THE TABLE
1565  006054  012701  001304                    MOV     #TUMTAB-2,%1
1566  006060  005721          8$:               TST     (1)+            ;INCREMENT POINTER
1567  006062  010046                            MOV     %0,-(6)         ;SAVE REGISTER ZERO
1568  006064  013700  016176                    MOV     LINE,%0         ;FETCH LINE NUMBER
1569  006070  005760  001146                    TST     WCT(0)          ;HAS THE LAST CHARACTER BEEN TRANSMITTED
1570  006074  001402                            BEQ     .+6             ;LAST CHARACTER HAS BEEN TRANSMITTED
1571  006076  012600                            MOV     (6)+,%0         ;RESTORE REGISTER ZERO
1572  006100  000674                            BR      2$              ;GO WAIT FOR NEXT CHARACTER
1573  006102  012600                            MOV     (6)+,%0         ;RESTORE REGISTER ZERO
1574  006104  012701  016350          9$:       MOV     #OUTBUF,%1
1575  006110  012702  016514                    MOV     #INBUF,%2
1576  006114  012705  000014                    MOV     #12.,%5         ;ROTATE PARITY BUFFER
1577  006120  004537  006250                    JSR     5,RORPARBUF     ;12. PLACES RIGHT
1578  006124  005037  001300          10$:      CLR     RCVDAT
1579  006130  005037  001302                    CLR     XMTDAT
1580  006134  020127  016513                    CMP     %1,#OUTBUF+99.  ;HAVE ALL CHARS. BEEN COMPARED
1581  006140  001441                            BEQ     15$
1582  006142  112137  001302                    MOVB    (1)+,XMTDAT     ;GET TRANSMITTED CHARACTER
1583  006146  043737  001304  001302            BIC     CARMSK,XMITDAT  ;CLEAR NON-TRANSMITTED BITS
1584  006154  111237  001300                    MOVB    (2),RCVDAT      ;GET RECEIVED CHARACTER
1585  006160  104002                            DATCHK                  ;COMPARE TRANS. & RCVD. CHARS.
1586
1587                                    ;ROUTINE TO COMPUTE AND CHECK PARITY ON RECEIVED DATA
1588  006162  012703  000010          11$:      MOV     #8.,%3          ;GET BIT COUNTER
1589  006166  005000                            CLR     %0              ;CLEAR COMPUTED PARITY INDICATOR
1590  006170  106037  001300          12$:      RORB    RCVDAT          ;LOOK AT RECEIVED BIT
```

```
1591  006174  103001                  BCC     13$             ;BRANCH IF A 0
1592  006176  005100                  COM     %0              ;COMPLEMENT R0 IF A 1
1593  006200  005303          13$:    DEC     %3              ;DECREMENT BIT COUNTER
1594  006202  001372                  BNE     12$             ;LOOK AT NEXT BIT IF NOT DONE
1595  006204  000240                  NOP                     ;IF COMPUTED PARITY WAS ODD R0 WILL
1596                                  ;CONTAIN ALL 1'S, IF EVEN R0 = 0
1597  006206  112237  001300          MOVB    (2)+,RCVDAT     ;GET RECEIVED CHARACTER
1598  006212  012705  000001          MOV     #1,%5           ;ROTATE PARITY BUFFER 1 PLACE
1599  006216  004537  006250          JSR     5,RORPARBUF     ;RIGHT LEAVING RECEIVED PARITY BIT IN CARRY
1600  006222  103004                  BCC     14$             ;BRANCH IF RECEIVED PARITY WAS EVEN
1601  006224  005700                  TST     %0              ;TEST FOR COMPUTED ODD PARITY
1602  006226  001336                  BNE     10$             ;BRANCH IF COMPUTED & RECEIVED WAS ODD
1603  006230  104001                  ERROR                   ;ERROR! COMPUTED =EVEN,RECEIVED = ODD
1604  006232  000734                  BR      10$             ;CONTINUE TEST
1605  006234  005700          14$:    TST     %0              ;TEST FOR EVEN COMPUTED PARITY
1606  006236  001732                  BEQ     10$             ;BRANCH IF COMPUTED PARITY WAS EVEN
1607  006240  104001                  ERROR                   ;ERROR! COMPUTED =ODD,RECEIVED = EVEN
1608  006242  000730                  BR      10$             ;CONTINUE TEST
1609  006244  005726          15$:    POPSP                   ;REPOSITION STACK POINTER
1610  006246  104006                  SCOPE                   ;SCOPE
1611
1612                                  ;ROUTINE TO ROTATE PARITY BUFFER.
1613  006250  006037  006312  RORPARBUF:ROR   PAR0
1614  006254  006037  006314          ROR     PAR1
1615  006260  006037  006316          ROR     PAR2
1616  006264  006037  006320          ROR     PAR3
1617  006270  006037  006322          ROR     PAR4
1618  006274  006037  006324          ROR     PAR5
1619  006300  006037  006326          ROR     PAR6
1620  006304  005316                  DEC     (SP)            ;DECREMENT ROTATE COUNT
1621  006306  001360                  BNE     RORPARBUF
1622  006310  000205                  RTS     5
1623                                  ;PARITY BUFFER
1624  006312  000000          PAR0:   OPEN
1625  006314  000000          PAR1:   OPEN
1626  006316  000000          PAR2:   OPEN
1627  006320  000000          PAR3:   OPEN
1628  006322  000000          PAR4:   OPEN
1629  006324  000000          PAR5:   OPEN
1630  006326  000000          PAR6:   OPEN
1631
1632                                  ;SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE
1633  006330  010046          GTLINB: MOV     %0,-(SP)        ;SAVE R0 ON THE STACK
1634  006332  005037  001272          CLR     @#LINBIT        ;CLEAR LINE BIT
1635  006336  013700  016176          MOV     @#LINE,%0       ;GET LINE
1636  006342  000261                  SEC                     ;SET CARRY
1637  006344  006137  001272  1$:     ROL     LINBIT          ;SHIFT LINE BIT
1638  006350  162700  000002          SUB     #2,%0           ;SUBTRACT 2 FROM LINE NUMBER
1639  006354  100373                  BPL     1$              ;BRANCH IF GREATER THAN 0
1640  006356  012600                  MOV     (SP)+,%0        ;RESTORE R0
1641  006360  000207                  RTS     7               ;EXIT SUBROUTINE
1642
```

```
                                PRGO:   TYPE
                                        PRGOM
006362  104000                  PRGOR:  MOV     #RT0,KSTART     ;GET ADDRESS OF FIRST TEST
006364  017427
006366  012737  006422  001506        CLR     RTNNO           ;CLEAR ROUTINE #
006374  005037  001512                JMP     SRSET
006400  000137  002240
006404  012737  006422  001506        MOV     #RT0,KSTART     ;GET ADDRESS OF FIRST TEST
006412  005037  001512                CLR     RTNNO           ;CLEAR ROUTINE NUMBER
006416  000137  002262                JMP     GETRDY          ;GO AND START PROGRAM
                        ;*****************************************************************************
006422  000000                RT0:    0                       ;ROUTINE # 0                            *
006424  006464                        RT1                     ;ADDR OF NEXT ROUTINE.                  *
006426  000144                        100.                    ;ITERATION COUNT                        *
006430  006432                        RT0A                    ;SCOPE ENTRY POINT.                     *
                                        X=X+1
                        ;*****************************************************************************

                        ;TEST ABILITY TO REFERENCE CSR WITHOUT TRAPPING
006432  012737  006454  000004  RT0A:   MOV     #1$,@#ERRVEC    ;SET UP ERROR TRAP.
006440  005777  172604                TST     @CSR            ;REFERENCE CSR
006444  012737  000006  000004        MOV     #ERRVEC+2,@#ERRVEC      ;RESET TIME OUT TRAP
006452  104006                        SCOPE
006454  162716  000004          1$:   SUB     #4,(6)          ;RESTORE PC TO WHERE THE ILLEGAL
                                                              ;REFERENCE OCCURED
006460  104001                        ERROR                   ;ERROR! ILLEGAL REFERENCE OCCURED
006462  000002                        RTI                     ;LOOP ILLEGAL REFERENCE INSTRUCTION
                        ;*****************************************************************************
000001                  RT1:    1                               ;ROUTINE # 1                            *
006466  006536                        RT2                     ;ADDR OF NEXT ROUTINE.                  *
006470  000144                        100.                    ;ITERATION COUNT                        *
006472  006474                        RT1A                    ;SCOPE ENTRY POINT.                     *
                                        X=X+1
                        ;*****************************************************************************

                        ;TEST THAT CSR BIT0 CAN BE SET AND CLEARED
006474  012777  000001  172546  RT1A:   MOV     #BIT0,@CSR      ;SET BIT0.
006502  022777  000001  172540        CMP     #BIT0,@CSR      ;TEST THAT BIT0 IS SET
006510  001402                        BEQ     1$              ;BRANCH IF SET
006512  104001                        ERROR                   ;CSR BIT0 FAILED TO SET
006514  000407                        BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
006516  042777  000001  172524  1$:   BIC     #BIT0,@CSR      ;CLEAR BIT0
006524  005777  172520                TST     @CSR            ;TEST THAT BIT0 IS CLEAR
006530  001401                        BEQ     2$
006532  104001                        ERROR                   ;CSR BIT0 FAILED TO CLEAR
006534  104006                  2$:   SCOPE
                        ;*****************************************************************************
006536  000002                RT2:    2                       ;ROUTINE # 2                            *
006540  006610                        RT3                     ;ADDR OF NEXT ROUTINE.                  *
006542  000144                        100.                    ;ITERATION COUNT                        *
006544  006546                        RT2A                    ;SCOPE ENTRY POINT.                     *
000002                                  X=X+1
                        ;*****************************************************************************

                        ;TEST THAT CSR BIT1 CAN BE SET AND CLEARED
006546  012777  000002  172474  RT2A:   MOV     #BIT1,@CSR      ;SET BIT1.
006554  022777  000002  172466        CMP     #BIT1,@CSR      ;TEST THAT BIT1 IS SET
```

```
1699  006562  001402                          BEQ     1$              ;BRANCH IF SET
1700  006564  104001                          ERROR                   ;CSR BIT1 FAILED TO SET
1701  006566  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1702  006570  042777  000002  172452  1$:     BIC     #BIT1,@CSR      ;CLEAR BIT1
1703  006576  005777  172446                  TST     @CSR            ;TEST THAT BIT1 IS CLEAR
1704  006602  001401                          BEQ     2$
1705  006604  104001                          ERROR                   ;CSR BIT1 FAILED TO CLEAR
1706  006606  104006                  2$:     SCOPE
1707                                  ;*****************************************************************************
1708  006610  000003                  RT3:    3                       ;ROUTINE # 3                                *
1709  006612  006662                          RT4                     ;ADDR OF NEXT ROUTINE.                      *
1710  006614  000144                          100.                    ;ITERATION COUNT                            *
1711  006616  006620                          RT3A                    ;SCOPE ENTRY POINT.                         *
1712           000003                         X=X+1
1713                                  ;*****************************************************************************
1714
1715                                  ;TEST THAT CSR BIT2 CAN BE SET AND CLEARED
1716  006620  012777  000004  172422  RT3A:   MOV     #BIT2,@CSR      ;SET BIT2.
1717  006626  022777  000004  172414          CMP     #BIT2,@CSR      ;TEST THAT BIT2 IS SET
1718  006634  001402                          BEQ     1$              ;BRANCH IF SET
1719  006636  104001                          ERROR                   ;CSR BIT2 FAILED TO SET
1720  006640  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1721  006642  042777  000004  172400  1$:     BIC     #BIT2,@CSR      ;CLEAR BIT2
1722  006650  005777  172374                  TST     @CSR            ;TEST THAT BIT2 IS CLEAR
1723  006654  001401                          BEQ     2$
1724  006656  104001                          ERROR                   ;CSR BIT2 FAILED TO CLEAR
1725  006660  104006                  2$:     SCOPE
1726                                  ;*****************************************************************************
1727  006662  000004                  RT4:    4                       ;ROUTINE # 4                                *
1728  006664  006734                          RT5                     ;ADDR OF NEXT ROUTINE.                      *
1729  006666  000144                          100.                    ;ITERATION COUNT                            *
1730  006670  006672                          RT4A                    ;SCOPE ENTRY POINT.                         *
1731           000004                         X=X+1
1732                                  ;*****************************************************************************
1733
1734                                  ;TEST THAT CSR BIT4 CAN BE SET AND CLEARED
1735  006672  012777  000020  172350  RT4A:   MOV     #BIT4,@CSR      ;SET BIT4.
1736  006700  022777  000020  172342          CMP     #BIT4,@CSR      ;TEST THAT BIT4 IS SET
1737  006706  001402                          BEQ     1$              ;BRANCH IF SET
1738  006710  104001                          ERROR                   ;CSR BIT4 FAILED TO SET
1739  006712  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1740  006714  042777  000020  172326  1$:     BIC     #BIT4,@CSR      ;CLEAR BIT4
1741  006722  005777  172322                  TST     @CSR            ;TEST THAT BIT4 IS CLEAR
1742  006726  001401                          BEQ     2$
1743  006730  104001                          ERROR                   ;CSR BIT4 FAILED TO CLEAR
1744  006732  104006                  2$:     SCOPE
1745                                  ;*****************************************************************************
1746  006734  000005                  RT5:    5                       ;ROUTINE # 5                                *
1747  006736  007006                          RT6                     ;ADDR OF NEXT ROUTINE.                      *
1748  006740  000144                          100.                    ;ITERATION COUNT                            *
1749  006742  006744                          RT5A                    ;SCOPE ENTRY POINT.                         *
1750           000005                         X=X+1
1751                                  ;*****************************************************************************
1752
1753                                  ;TEST THAT CSR BIT5 CAN BE SET AND CLEARED
1754  006744  012777  000040  172276  RT5A:   MOV     #BIT5,@CSR      ;SET BIT5.
```

```
1755  006752  022777  000040  172270          CMP     #BIT5,@CSR      ;TEST THAT BIT5 IS SET
1756  006760  001402                          BEQ     1$              ;BRANCH IF SET
1757  006762  104001                          ERROR                   ;CSR BIT5 FAILED TO SET
1758  006764  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1759  006766  042777  000040  172254  1$:     BIC     #BIT5,@CSR      ;CLEAR BIT5
1760  006774  005777  172250                  TST     @CSR            ;TEST THAT BIT5 IS CLEAR
1761  007000  001401                          BEQ     2$
1762  007002  104001                          ERROR                   ;CSR BIT5 FAILED TO CLEAR
1763  007004  104006                  2$:     SCOPE
1764                                  ;*****************************************************************************
1765  007006  000006                  RT6:    6                       ;ROUTINE # 6                                *
1766  007010  007060                          RT7                     ;ADDR OF NEXT ROUTINE.                      *
1767  007012  000144                          100.                    ;ITERATION COUNT                            *
1768  007014  007016                          RT6A                    ;SCOPE ENTRY POINT.                         *
1769          000006                          X=X+1
1770                                  ;*****************************************************************************
1771
1772                                  ;TEST THAT CSR BIT6 CAN BE SET AND CLEARED
1773  007016  012777  000100  172224  RT6A:   MOV     #BIT6,@CSR      ;SET BIT6.
1774  007024  022777  000100  172216          CMP     #BIT6,@CSR      ;TEST THAT BIT6 IS SET
1775  007032  001402                          BEQ     1$              ;BRANCH IF SET
1776  007034  104001                          ERROR                   ;CSR BIT6 FAILED TO SET
1777  007036  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1778  007040  042777  000100  172202  1$:     BIC     #BIT6,@CSR      ;CLEAR BIT6
1779  007046  005777  172176                  TST     @CSR            ;TEST THAT BIT6 IS CLEAR
1780  007052  001401                          BEQ     2$
1781  007054  104001                          ERROR                   ;CSR BIT6 FAILED TO CLEAR
1782  007056  104006                  2$:     SCOPE
1783                                  ;*****************************************************************************
1784  007060  000007                  RT7:    7                       ;ROUTINE # 7                                *
1785  007062  007132                          RT10                    ;ADDR OF NEXT ROUTINE.                      *
1786  007064  000144                          100.                    ;ITERATION COUNT                            *
1787  007066  007070                          RT7A                    ;SCOPE ENTRY POINT.                         *
1788          000007                          X=X+1
1789                                  ;*****************************************************************************
1790
1791                                  ;TEST THAT CSR BIT12 CAN BE SET AND CLEARED
1792  007070  012777  010000  172152  RT7A:   MOV     #BIT12,@CSR     ;SET BIT12.
1793  007076  022777  010000  172144          CMP     #BIT12,@CSR     ;TEST THAT BIT12 IS SET
1794  007104  001402                          BEQ     1$              ;BRANCH IF SET
1795  007106  104001                          ERROR                   ;CSR BIT12 FAILED TO SET
1796  007110  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1797  007112  042777  010000  172130  1$:     BIC     #BIT12,@CSR     ;CLEAR BIT12
1798  007120  005777  172124                  TST     @CSR            ;TEST THAT BIT12 IS CLEAR
1799  007124  001401                          BEQ     2$
1800  007126  104001                          ERROR                   ;CSR BIT12 FAILED TO CLEAR
1801  007130  104006                  2$:     SCOPE
1802                                  ;*****************************************************************************
1803  007132  000010                  RT10:   10                      ;ROUTINE # 10                               *
1804  007134  007204                          RT11                    ;ADDR OF NEXT ROUTINE.                      *
1805  007136  000144                          100.                    ;ITERATION COUNT                            *
1806  007140  007142                          RT10A                   ;SCOPE ENTRY POINT.                         *
1807          000010                          X=X+1
1808                                  ;*****************************************************************************
1809
1810                                  ;TEST THAT CSR BIT13 CAN BE SET AND CLEARED
```

```
1811  007142  012777  020000  172100  RT10A:  MOV     #BIT13,@CSR     ;SET BIT13.
1812  007150  022777  020000  172072          CMP     #BIT13,@CSR     ;TEST THAT BIT13 IS SET
1813  007156  001402                          BEQ     1$              ;BRANCH IF SET
1814  007160  104001                          ERROR                   ;CSR BIT13 FAILED TO SET
1815  007162  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1816  007164  042777  020000  172056  1$:     BIC     #BIT13,@CSR     ;CLEAR BIT13
1817  007172  005777  172052                  TST     @CSR            ;TEST THAT BIT13 IS CLEAR
1818  007176  001401                          BEQ     2$
1819  007200  104001                          ERROR                   ;CSR BIT13 FAILED TO CLEAR
1820  007202  104006                  2$:     SCOPE
1821                                  ;*************************************************************************
1822  007204  000011                  RT11:   11                      ;ROUTINE # 11                          *
1823  007206  007274                          RT12                    ;ADDR OF NEXT ROUTINE.                 *
1824  007210  000144                          100.                    ;ITERATION COUNT                       *
1825  007212  007214                          RT11A                   ;SCOPE ENTRY POINT.                    *
1826          000011                          X=X+1
1827                                  ;*************************************************************************
1828
1829                                  ;TEST THAT RESET & CLEAR INSTRUCTION CLEAR ALL R/W BITS IN THE CONTROL
1830                                  ; STATUS REG. (CSR)
1831  007214  012777  030167  172026  RT11A:  MOV     #30167,@CSR     ;SET ALL R/W BITS IN THE CSR
1832  007222  005037  001302                  CLR     XMTDAT
1833  007226  104005                          SRESET                  ;ISSUE RESET
1834  007230  017737  172014  001300          MOV     @CSR,RCVDAT     ;GET CSR CONTENTS
1835  007236  001402                          BEQ     1$              ;BRANCH IF RESET CLEARED ALL BITS
1836  007240  104011                          ERROR1                  ;ERROR! RESET DID NOT CLEAR ALL BITS
1837  007242  000764                          BR      RT11A           ;LOOP ON ERROR
1838  007244  012777  030167  171776  1$:     MOV     #30167,@CSR     ;SET ALL R/W BITS IN CSR
1839  007252  005077  171772                  CLR     @CSR            ;CLEAR THE CSR
1840  007256  017737  171766  001300          MOV     @CSR,RCVDAT     ;GET & TEST CSR
1841  007264  001402                          BEQ     2$              ;GO TO EXIT IF RESULT = 0
1842  007266  104011                          ERROR1                  ;ERROR! CLEAR INST. DID NOT CLEAR ALL BITS
1843  007270  000765                          BR      1$              ;LOOP ERROR
1844  007272  104006                  2$:     SCOPE                   ;SCOPE
1845                                  ;*************************************************************************
1846  007274  000012                  RT12:   12                      ;ROUTINE # 12                          *
1847  007276  007430                          RT13                    ;ADDR OF NEXT ROUTINE.                 *
1848  007300  000012                          10.                     ;ITERATION COUNT                       *
1849  007302  007304                          RT12A                   ;SCOPE ENTRY POINT.                    *
1850          000012                          X=X+1
1851                                  ;*************************************************************************
1852
1853                                  ;TEST THAT A BINARY COUNT CAN BE LOADED INTO A CLEAR BKCSR AND THAT
1854                                  ;A BINARY COUNT CAN BE CLEARED.
1855  007304  005037  001302          RT12A:  CLR     XMTDAT
1856  007310  013777  001302  171736  1$:     MOV     XMTDAT,@BKCSR   ;LOAD BINARY COUNT INTO BKCSR
1857  007316  017737  171732  001300          MOV     @BKCSR,RCVDAT   ;GET BKCSR DATA
1858  007324  023737  001302  001300          CMP     XMTDAT,RCVDAT   ;COMPARE DATA LOADED & DATA READ BACK
1859  007332  001405                          BEQ     2$              ;BRANCH IF DATA COMPARES
1860  007334  104011                          ERROR1                  ;ERROR! DATA DOD NIT COMPARE
1861  007336  032777  040000  171536          BIT     #BIT14,@SWR     ;SCOPE LOOP?
1862  007344  001361                          BNE     1$              ;BRANCH IF SCOPE LOOP
1863  007346  013701  001302          2$:     MOV     XMTDAT,%1       ;SAVE BINARY COUNT
1864  007352  005037  001302                  CLR     XMTDAT
1865  007356  005077  171672          3$:     CLR     @BKCSR          ;CLEAR BKCSR AND TEST
1866  007362  017737  171666  001300          MOV     @BKCSR,RCVDAT   ;BKCSR CAN BE CLEARED
```

```
1867  007370  001405                        BEQ     4$              ;BRANCH IF BKCSR CLEARED
1868  007372  104011                        ERROR1                  ;ERROR! BKCSR DID NOT CLEAR
1869  007374  032777  040000  171500        BIT     #BIT14,@SWR     ;SCOPE LOOP?
1870  007402  001365                        BNE     3$              ;BRANCH IF SCOPE LOOP
1871  007404  010137  001302          4$:   MOV     %1,XMTDAT       ;GET BINARY COUNT
1872  007410  023727  001302  177777        CMP     XMTDAT,#-1      ;ALL NUMBERS BEEN LOADED
1873  007416  001403                        BEQ     5$              ;GO TO EXIT
1874  007420  005237  001302                INC     XMTDAT          ;INCREMENT BINARY COUNT
1875  007424  000731                        BR      1$              ;REPEAT TEST
1876  007426  104006                  5$:   SCOPE                   ;SCOPE
1877                                  ;*****************************************************************************
1878  007430  000013                  RT13:   13                    ;ROUTINE # 13                                 *
1879  007432  007542                          RT14                  ;ADDR OF NEXT ROUTINE.                        *
1880  007434  000144                          100.                  ;ITERATION COUNT                              *
1881  007436  007440                          RT13A                 ;SCOPE ENTRY POINT.                           *
1882          000013                          X=X+1
1883                                  ;*****************************************************************************
1884
1885                                  ;TEST THAT RANDOM NUMBERS CAN BE LOADED INTO THE BKCSR
1886  007440  012702  010000          RT13A:  MOV     #10000,%2       ;GET RANDOM #COUNTER
1887  007444  017737  171604  001616  1$:     MOV     @BKCSR,PRVCNT   ;GET PREVIOUS CONTENTS
1888  007452  004737  002712                  JSR     7,@#RNGEN       ;GO GET A RANDOM NUMBER
1889  007456  010037  001302                  MOV     %0,XMTDAT       ;GET RANDOM NUMBER
1890  007462  013777  001302  171564  2$:     MOV     XMTDAT,@BKCSR   ;LOAD RANDOM NUMBER INTO BKCSR
1891  007470  017737  171560  001300          MOV     @BKCSR,RCVDAT   ;GET BKCSR DATA
1892  007476  023737  001302  001300          CMP     XMTDAT,RCVDAT   ;COMPARE DATA
1893  007504  001401                        BEQ     3$              ;BRANCH IF SAME
1894  007506  104011                        ERROR1                  ;ERROR! DATA NOT THE SAME
1895  007510  032777  040000  171364  3$:     BIT     #BIT14,@SWR     ;SCOPE LOOP?
1896  007516  001406                        BEQ     4$              ;BRANCH IF NO LOOP ON ERROR
1897  007520  005077  171530                  CLR     @BKCSR          ;
1898  007524  013777  001615  171522          MOV     PRVCNT,@BKCSR   ;LOAD PREVIOUS CONTENTS
1899  007532  000753                        BR      2$              ;REPEAT TEST
1900  007534  005302                  4$:   DEC     %2
1901  007536  001342                        BNE     1$              ;BRANCH IF NOT
1902  007540  104006                  5$:   SCOPE                   ;SCOPE
1903                                  ;*****************************************************************************
1904  007542  000014                  RT14:   14                    ;ROUTINE # 14                                 *
1905  007544  007602                          RT15                  ;ADDR OF NEXT ROUTINE.                        *
1906  007546  000012                          10.                   ;ITERATION COUNT                              *
1907  007550  007552                          RT14A                 ;SCOPE ENTRY POINT.                           *
1908          000014                          X=X+1
1909                                  ;*****************************************************************************
1910
1911                                  ;TEST THAT RESET CLEARS ALL BREAK STATUS REGISTER BITS
1912
1913  007552  012777  177777  171474  RT14A:  MOV     #-1,@BKCSR
1914  007560  005037  001302                  CLR     XMTDAT
1915  007564  104005                          RESET
1916  007566  017737  171462  001300          MOV     @BKCSR,RCVDAT
1917  007574  001401                        BEQ     1$
1918  007576  104011                        ERROR1
1919  007600  104006                  1$:   SCOPE
1920                                  ;*****************************************************************************
1921  007602  000015                  RT15:   15                    ;ROUTINE # 15                                 *
1922  007604  007736                          RT16                  ;ADDR OF NEXT ROUTINE.                        *
```

```
1923  007606  000012                          10.                     ;ITERATION COUNT                        *
1924  007610  007612                          RT15A                   ;SCOPE ENTRY POINT.                     *
1925          000015                          X=X+1
1926                                  ;*************************************************************************
1927
1928                                  ;TEST THAT A BINARY COUNT CAN BE LOADED INTO A CLEAR BASREG AND THAT
1929                                  ;A BINARY COUNT CAN BE CLEARED.
1930  007612  005037  001302          RT15A:  CLR     XMTDAT
1931  007616  013777  001302  171432  1$:     MOV     XMTDAT,@BASREG  ;LOAD BINARY COUNT INTO BASREG
1932  007624  017737  171426  001300          MOV     @BASREG,RCVDAT  ;GET BASREG DATA
1933  007632  023737  001302  001300          CMP     XMTDAT,RCVDAT   ;COMPARE DATA
1934  007640  001405                          BEQ     2$              ;BRANCH IF DATA COMPARES
1935  007642  104011                          ERROR1                  ;ERROR! DATA DID NOT COMPARE
1936  007644  032777  040000  171230          BIT     #BIT14,@SWR     ;SCOPE LOOP?
1937  007652  001361                          BNE     1$              ;BRANCH IF SCOPE LOOP
1938  007654  013701  001302          2$:     MOV     XMTDAT,%1       ;SAVE BINARY COUNT
1939  007660  005037  001302                  CLR     XMTDAT
1940  007664  005077  171366          3$:     CLR     @BASREG
1941  007670  017737  171362  001300          MOV     @BASREG,RCVDAT
1942  007676  001405                          BEQ     4$              ;BRANCH IF BKCSR CLEARED
1943  007700  104011                          ERROR1                  ;ERROR! BKCSR DID NOT CLEAR
1944  007702  032777  040000  171172          BIT     #BIT14,@SWR     ;SCOPE LOOP?
1945  007710  001365                          BNE     3$              ;BRANCH IF SCOPE LOOP
1946  007712  010137  001302          4$:     MOV     %1,XMTDAT       ;GET BINARY COUNT
1947  007716  023727  001302  177000          CMP     XMTDAT,#177000  ;ALL NUMBERS BEEN LOADED
1948  007724  001403                          BEQ     5$              ;GO TO EXIT
1949  007726  105237  001303                  INCB    XMTDAT+1        ;INCREMENT BINARY COUNT
1950  007732  000731                          BR      1$              ;REPEAT TEST
1951  007734  104006                  5$:     SCOPE                   ;SCOPE
1952                                  ;*************************************************************************
1953  007736  000016                  RT16:   16                      ;ROUTINE # 16                           *
1954  007740  010054                          RT17                    ;ADDR OF NEXT ROUTINE.                  *
1955  007742  000144                          100.                    ;ITERATION COUNT                        *
1956  007744  007746                          RT16A                   ;SCOPE ENTRY POINT.                     *
1957          000016                          X=X+1
1958                                  ;*************************************************************************
1959
1960                                  ;TEST THAT RANDOM NUMBERS CAN BE LOADED INTO THE BASE REGISTER
1961  007746  012702  010000          RT16A:  MOV     #10000,%2       ;GET RANDOM #COUNTER
1962  007752  017737  171300  001616  1$:     MOV     @BASREG,PRVCNT  ;GET PREVIOUS CONTENTS
1963  007760  004737  002712                  JSR     7,@#RNGEN       ;GO GET A RANDOM NUMBER
1964  007764  042700  000377                  BIC     #000377,%0      ;CLEAR UNUSED BITS
1965  007770  010037  001302                  MOV     %0,XMTDAT       ;GET RANDOM NUMBER
1966  007774  013777  001302  171254  2$:     MOV     XMTDAT,@BASREG  ;LOAD RANDOM NUMBER INTO BASREG
1967  010002  017737  171250  001300          MOV     @BASREG,RCVDAT  ;GET BASREG DATA
1968  010010  023737  001302  001300          CMP     XMTDAT,RCVDAT   ;COMPARE DATA
1969  010016  001401                          BEQ     3$              ;BRANCH IF SAME
1970  010020  104011                          ERROR1                  ;ERROR! DATA NOT THE SAME
1971  010022  032777  040000  171052  3$:     BIT     #BIT14,@SWR     ;SCOPE LOOP?
1972  010030  001406                          BEQ     4$              ;BRANCH IF NO LOOP ON ERROR
1973  010032  005077  171220                  CLR     @BASREG
1974  010036  013777  001616  171212          MOV     PRVCNT,@BASREG  ;LOAD PREVIOUS CONTENTS
1975  010044  000753                          BR      2$              ;REPEAT TEST
1976  010046  005302                  4$:     DEC     %2              ;10000 NUMBERS BEEN TESTED
1977  010050  001340                          BNE     1$              ;BRANCH IF NOT
1978  010052  104006                  5$:     SCOPE                   ;SCOPE
```

```
 1979
 1980  010054  000017                  ;*********************************************************************
 1981  010056  010146                  RT17:   17                      ;ROUTINE # 17                        *
 1982  010060  000144                          RT20                    ;ADDR OF NEXT ROUTINE.               *
 1983  010062  010064                          100.                    ;ITERATION COUNT                     *
 1984          000017                          RT17A                   ;SCOPE ENTRY POINT.                  *
 1985                                          X=X+1                   ;*********************************************************************
 1986
 1987                                  ;TEST THAT ALL BAR BITS CAN BE INDIVIDUALLY SET AND CLEARED
 1988  010064  013701  001252          RT17A:  MOV     BAR,%1          ;GET BAR ADDRESS
 1989  010070  012777  001106  171160          MOV     #CAT,@BASREG    ;INITIALIZE BASE REGISTER
 1990  010076  012700  000001                  MOV     #1,%0           ;GET BIT TESTER
 1991  010102  050011                  1$:     BIS     %0,(1)          ;SET BAR BIT
 1992  010104  020011                          CMP     %0,(1)          ;TEST THAT ONLY THE PROPER BAR BIT SET
 1993  010106  001006                          BNE     3$              ;BRANCH IF ERROR
 1994  010110  040011                          BIC     %0,(1)          ;CLEAR BAR BIT
 1995  010112  005711                          TST     (1)             ;TEST THAT BAR BIT CLEARED
 1996  010114  001[illegible]                  BNE     5$              ;BRANCH IF BAR BIT FAILED TO CLEAR
 1997  010116  0[illegible]                    ASL     %0              ;SHIFT BIT TESTER
 1998  010120  1[illegible]                    BCC     1$
 1999  010122  104006                  2$:     SCOPE                   ;SCOPE
 2000  010124  010037  001302          3$:     MOV     %0,XMTDAT       ;GET WHAT DATA WAS SUPPOSED TO BE
 2001  010130  011137  001300          4$:     MOV     (1),RCVDAT      ;GET WHAT DATA WAS
 2002  010134  104011                          ERROR1                  ;ERROR! IMPROPER BIT OPERATION
 2003  010136  000771                          BR      2$              ;GO TO SCOPE
 2004  010140  005037  001302          5$:     CLR     XMTDAT          ;GET WHAT DATA WAS SUPPOSED TO BE
 2005  010144  000771                          BR      4$
 2006                                  ;*********************************************************************
 2007  010146  000020                  RT20:   20                      ;ROUTINE # 20                        *
 2008  010150  010234                          RT21                    ;ADDR OF NEXT ROUTINE.               *
 2009  010152  000012                          10.                     ;ITERATION COUNT                     *
 2010  010154  010156                          RT20A                   ;SCOPE ENTRY POINT.                  *
 2011          000020                          X=X+1                   ;*********************************************************************
 2012
 2013
 2014                                  ;TEST THAT RESET CLEARS ALL BAR BITS
 2015  010156  005037  001302          RT20A:  CLR     XMTDAT
 2016  010162  052777  177777  171062          BIS     #-1,@BAR        ;SET ALL BAR BITS
 2017  010170  104005                          $RESET                  ;RESET
 2018  010172  017737  171054  001300          MOV     @BAR,RCVDAT     ;GET BAR DATA
 2019  010200  001402                          BEQ     1$              ;BRANCH IF ALL 0'S
 2020  010202  104011                          ERROR1                  ;ERROR! RESET DID NOT CLEAR ALL BAR BITS
 2021  010204  000412                          BR      2$              ;GO TO EXIT
 2022  010206  052777  177777  171036  1$:     BIS     #-1,@BAR        ;SET ALL BIT IN THE BAR
 2023  010214  005077  171032                  CLR     @BAR            ;CLEAR ALL BITS IN THE BAR
 2024  010220  017737  171026  001300          MOV     @BAR,RCVDAT     ;GET & TEST RESULT OF CLEAR OPERATION
 2025  010226  001401                          BEQ     2$              ;EXIT IF ALL BITS CLEARED
 2026  010230  104011                          ERROR1                  ;ERROR! ALL BITS DID NOT CLEAR
 2027  010232  104006                  2$:     SCOPE                   ;SCOPE
 2028                                  ;*********************************************************************
 2029  010234  000021                  RT21:   21                      ;ROUTINE # 21                        *
 2030  010236  010342                          RT22                    ;ADDR OF NEXT ROUTINE.               *
 2031  010240  000144                          100.                    ;ITERATION COUNT                     *
 2032  010242  010244                          RT21A                   ;SCOPE ENTRY POINT.                  *
 2033          000021                          X=X+1                   ;*********************************************************************
 2034
```

```
                                        ;TEST THAT CSR RESPONDS PROPERLY TO BYTE COMMANDS
010244  012777  010100  170776  RT21A:  MOV     #10100,@CSR     ;LOAD TEST NUMBER IN CSR
010252  105077  170772                  CLRB    @CSR            ;CLEAR EVEN BYTE
010256  022777  010000  170764          CMP     #10000,@CSR     ;TEST THAT ONLY EVEN BYTE CLEARED
010264  001410                          BEQ     1$
010266  012737  010[illegible]  001302  MOV     #10100,XMTDAT   ;LOAD CORRECT RESULT
010274  017737  170[illegible]  001300  MOV     @CSR,RCVDAT     ;GET ACTUAL RESULT
010302  104011                          ERROR1                  ;ERROR! EVEN BYTE INSTRUCTION FAILED
010304  000415                          BR      2$              ;GO TO SCOPE
010306  012777  010100  170734  1$:     MOV     #10100,@CSR     ;LOAD TEST NUMBER IN CSR
010314  105077  171302                  CLRB    @.CSR           ;TEST THAT ONLY ODD BYTE CLEARED
010320  001407                          BEQ     2$
010322  012737  000100  001302          MOV     #00100,XMTDAT   ;LOAD CORRECT RESULT
010330  017737  170714  001300          MOV     @CSR,RCVDAT     ;LOAD ACTUAL RESULT
010336  104011                          ERROR1                  ;ERROR! ODD BYTE INSTRUCTION FAILED
010340  104006                  2$:     SCOPE                   ;SCOPE
                                        ;*****************************************************************************
010342  000022                  RT22:   22                      ;ROUTINE # 22                                        *
010344  010450                          RT23                    ;ADDR OF NEXT ROUTINE.                               *
010346  000144                          100.                    ;ITERATION COUNT                                     *
010350  010352                          RT22A                   ;SCOPE ENTRY POINT.                                  *
        000022                          X=X+1
                                        ;*****************************************************************************

                                        ;TEST THAT BAR RESPONDS PROPERLY TO BYTE COMMANDS
010352  012777  010100  170672  RT22A:  MOV     #10100,@BAR     ;LOAD TEST NUMBER IN BAR
010360  105077  170666                  CLRB    @BAR            ;CLEAR EVEN BYTE
010364  022777  010000  170660          CMP     #10000,@BAR     ;TEST THAT ONLY EVEN BYTE CLEARED
010372  001410                          BEQ     1$
010374  012737  010100  001302          MOV     #10100,XMTDAT   ;LOAD CORRECT RESULT
010402  017737  170644  001300          MOV     @BAR,RCVDAT     ;GET ACTUAL RESULT
010410  104011                          ERROR1                  ;ERROR! EVEN BYTE INSTRUCTION FAILED
010412  000415                          BR      2$              ;GO TO SCOPE
010414  012777  010100  170630  1$:     MOV     #10100,@BAR     ;LOAD TEST NUMBER IN BAR
010422  105077  171176                  CLRB    @.BAR           ;TEST THAT ONLY ODD BYTE CLEARED
010426  001407                          BEQ     2$
010430  012737  000100  001302          MOV     #00100,XMTDAT   ;LOAD CORRECT RESULT
010436  017737  170606  001300          MOV     @CSR,RCVDAT     ;LOAD ACTUAL RESULT
010444  104011                          ERROR1                  ;ERROR! ODD BYTE INSTRUCTION FAILED
010446  104006                  2$:     SCOPE                   ;SCOPE
                                        ;*****************************************************************************
010450  000023                  RT23:   23                      ;ROUTINE # 23                                        *
010452  010556                          RT24                    ;ADDR OF NEXT ROUTINE.                               *
010454  000144                          100.                    ;ITERATION COUNT                                     *
010456  010460                          RT23A                   ;SCOPE ENTRY POINT.                                  *
        000023                          X=X+1
                                        ;*****************************************************************************

                                        ;TEST THAT BKCSR RESPONDS PROPERLY TO BYTE COMMANDS
010460  012777  010100  170566  RT23A:  MOV     #10100,@BKCSR   ;LOAD TEST NUMBER IN BKCSR
010466  105077  170562                  CLRB    @BKCSR          ;CLEAR EVEN BYTE
010472  022777  010000  170554          CMP     #10000,@BKCSR   ;TEST THAT ONLY EVEN BYTE CLEARED
010500  001410                          BEQ     1$
010502  012737  010100  001302          MOV     #10100,XMTDAT   ;LOAD CORRECT RESULT
010510  017737  170540  001300          MOV     @BKCSR,RCVDAT   ;GET ACTUAL RESULT
```

```
2091  010516  104011                          ERROR1                  ;ERROR! EVEN BYTE INSTRUCTION FAILED
2092  010520  000415                          BR      2$              ;GO TO SCOPE
2093  010522  012777  010100  170524  1$:     MOV     #10100,@BKCSR   ;LOAD TEST NUMBER IN BKCSR
2094  010530  105077  171072                  CLRB    @.BKCSR         ;TEST THAT ONLY ODD BYTE CLEARED
2095  010534  001407                          BEQ     2$
2096  010536  012737  000100  001302          MOV     #00100,XMTDAT   ;LOAD CORRECT RESULT
2097  010544  017737  170500  001300          MOV     @CSR,RCVDAT     ;LOAD ACTUAL RESULT
2098  010552  104011                          ERROR1                  ;ERROR! ODD BYTE INSTRUCTION FAILED
2099  010554  104006                  2$:     SCOPE                   ;SCOPE
2100                                  ;*****************************************************************************
2101  010556  000024                  RT24:   24                      ;ROUTINE # 24                              *
2102  010560  010622                          RT25                    ;ADDR OF NEXT ROUTINE.                     *
2103  010562  000144                          100.                    ;ITERATION COUNT                           *
2104  010564  010566                          RT24A                   ;SCOPE ENTRY POINT.                        *
2105           000024                         X=X+1
2106                                  ;*****************************************************************************
2107
2108                                  ;TEST THAT OVER RUN BIT (CSR BIT13) CAUSES AN INTERRUPT WHEN SET
2109  010566  012777  010620  170470  RT24A:  MOV     #1$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
2110  010574  012777  010000  170446          MOV     #BIT12,@CSR     ;SET TRANSMITTER IE BIT
2111  010602  052777  020000  170440          BIS     #BIT13,@CSR     ;SET OVER RUN BIT
2112  010610  005037  177776                  CLR     @#PSW           ;ENABLE INTERRUPTS
2113  010614  000240                          NOP
2114  010616  104001                          ERROR                   ;ERROR!OVERRUN FAILED TO CAUSE AN
2115                                                                  ;INTERRUPT,OR INTERRUPTED TO INCOR-
2116                                                                  ;RECT ADDRESS
2117  010620  104006                  1$:     SCOPE                   ;SCOPE
2118                                  ;*****************************************************************************
2119  010622  000025                  RT25:   25                      ;ROUTINE # 25                              *
2120  010624  010730                          RT26                    ;ADDR OF NEXT ROUTINE.                     *
2121  010626  000100                          100                     ;ITERATION COUNT                           *
2122  010630  010632                          RT25A                   ;SCOPE ENTRY POINT.                        *
2123           000025                         X=X+1
2124                                  ;*****************************************************************************
2125
2126                                  ;TEST THAT THE DM11 INTERRUPTS AT THE CORRECT LEVEL
2127  010632  012737  000340  177776  RT25A:  MOV     #PRTY7,@#PSW
2128  010640  012777  010670  170416          MOV     #1$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
2129  010646  012777  030000  170374          MOV     #30000,@CSR     ;SET OVER RUN & IE BITS
2130  010654  012737  000200  177776          MOV     #PRTY4,@#PSW    ;ALLOW INTERRUPTS ON LEVEL 5 & ABOVE
2131  010662  000240                          NOP
2132  010664  104001                          ERROR                   ;ERROR!DM11 FAILED TO INTERRUPT
2133  010666  000417                          BR      3$              ;GO TO EXIT
2134  010670  022626                  1$:     CMP     (6)+,(6)+       ;RESET STACK POINTER
2135  010672  013737  001266  177776          MOV     XMTLVL,@#PSW    ;LOAD DM11 INTERRUPT LEVEL
2136  010700  012777  010724  170356          MOV     #2$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
2137  010706  005077  170336                  CLR     @CSR
2138  010712  012777  030000  170330          MOV     #30000,@CSR
2139  010720  000240                          NOP
2140  010722  000401                          BR      3$              ;GO TO EXIT
2141  010724  104001                  2$:     ERROR                   ;ERROR! DM11 INTERRUPTED ON HIGHER
2142                                                                  ;PRIORITY LEVEL THAN SET FOR
2143  010726  104006                  3$:     SCOPE
2144                                  ;*****************************************************************************
2145  010730  000026                  RT26:   26                      ;ROUTINE # 26                              *
2146  010732  010746                          RT27                    ;ADDRESS OF NEXT TEST.                     *
```

```
2147  010734  000144                    100.                    ;ITERATION COUNT                        *
2148  010736  010740                    LTSTO                   ;SCOPE ENTRY POINT                      *
2149          000026                    X=X+1
2150                                ;*****************************************************************************
2151                                ;TRANSMITTER LINE TEST LINE 0
2152  010740  004537  004106      LTSTO:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 0
2153  010744  000000                    LINE0
2154          000001                    Y=Y+1
2155                                ;*****************************************************************************
2156  010746  000027                RT27:   27                      ;ROUTINE # 27                           *
2157  010750  010764                    RT30                    ;ADDRESS OF NEXT TEST.                  *
2158  010752  000144                    100.                    ;ITERATION COUNT                        *
2159  010754  010756                    LTST1                   ;SCOPE ENTRY POINT                      *
2160          000027                    X=X+1
2161                                ;*****************************************************************************
2162                                ;TRANSMITTER LINE TEST LINE 1
2163  010756  004537  004106      LTST1:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 1
2164  010762  000002                    LINE1
2165          000002                    Y=Y+1
2166                                ;*****************************************************************************
2167  010764  000030                RT30:   30                      ;ROUTINE # 30                           *
2168  010766  011002                    RT31                    ;ADDRESS OF NEXT TEST.                  *
2169  010770  000144                    100.                    ;ITERATION COUNT                        *
2170  010772  010774                    LTST2                   ;SCOPE ENTRY POINT                      *
2171          000030                    X=X+1
2172                                ;*****************************************************************************
2173                                ;TRANSMITTER LINE TEST LINE 2
2174  010774  004537  004106      LTST2:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 2
2175  011000  000004                    LINE2
2176          000003                    Y=Y+1
2177                                ;*****************************************************************************
2178  011002  000031                RT31:   31                      ;ROUTINE # 31                           *
2179  011004  011020                    RT32                    ;ADDRESS OF NEXT TEST.                  *
2180  011006  000144                    100.                    ;ITERATION COUNT                        *
2181  011010  011012                    LTST3                   ;SCOPE ENTRY POINT                      *
2182          000031                    X=X+1
2183                                ;*****************************************************************************
2184                                ;TRANSMITTER LINE TEST LINE 3
2185  011012  004537  004106      LTST3:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 3
2186  011016  000006                    LINE3
2187          000004                    Y=Y+1
2188                                ;*****************************************************************************
2189  011020  000032                RT32:   32                      ;ROUTINE # 32                           *
2190  011022  011036                    RT33                    ;ADDRESS OF NEXT TEST.                  *
2191  011024  000144                    100.                    ;ITERATION COUNT                        *
2192  011026  011030                    LTST4                   ;SCOPE ENTRY POINT                      *
2193          000032                    X=X+1
2194                                ;*****************************************************************************
2195                                ;TRANSMITTER LINE TEST LINE 4
2196  011030  004537  004106      LTST4:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 4
2197  011034  000010                    LINE4
2198          000005                    Y=Y+1
2199                                ;*****************************************************************************
2200  011036  000033                RT33:   33                      ;ROUTINE # 33                           *
2201  011040  011054                    RT34                    ;ADDRESS OF NEXT TEST.                  *
2202  011042  000144                    100.                    ;ITERATION COUNT                        *
```

```
2203  011044  011046                        LTST5                   ;SCOPE ENTRY POINT                      *
2204          000033                        X=X+1
2205                                ;*************************************************************************
2206                                ;TRANSMITTER LINE TEST LINE 5
2207  011046  004537  004106        LTST5:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 5
2208  011052  000012                        LINE5
2209          000006                        Y=Y+1
2210                                ;*************************************************************************
2211  011054  000034                RT34:   34                      ;ROUTINE # 34                           *
2212  011056  011072                        RT35                    ;ADDRESS OF NEXT TEST.                  *
2213  011060  000144                        100.                    ;ITERATION COUNT                        *
2214  011062  011064                        LTST6                   ;SCOPE ENTRY POINT                      *
2215          000034                        X=X+1
2216                                ;*************************************************************************
2217                                ;TRANSMITTER LINE TEST LINE 6
2218  011064  004537  004106        LTST6:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 6
2219  011070  000014                        LINE6
2220          000007                        Y=Y+1
2221                                ;*************************************************************************
2222  011072  000035                RT35:   35                      ;ROUTINE # 35                           *
2223  011074  011110                        RT36                    ;ADDRESS OF NEXT TEST.                  *
2224  011076  000144                        100.                    ;ITERATION COUNT                        *
2225  011100  011102                        LTST7                   ;SCOPE ENTRY POINT                      *
2226          000035                        X=X+1
2227                                ;*************************************************************************
2228                                ;TRANSMITTER LINE TEST LINE 7
2229  011102  004537  004106        LTST7:JSR       5,XMTTST        ;GO TEST TRANSMITTER LINE 7
2230  011106  000016                        LINE7
2231          000010                        Y=Y+1
2232                                ;*************************************************************************
2233  011110  000036                RT36:   36                      ;ROUTINE # 36                           *
2234  011112  011126                        RT37                    ;ADDRESS OF NEXT TEST.                  *
2235  011114  000144                        100.                    ;ITERATION COUNT                        *
2236  011116  011120                        LTST10                  ;SCOPE ENTRY POINT                      *
2237          000036                        X=X+1
2238                                ;*************************************************************************
2239                                ;TRANSMITTER LINE TEST LINE 10
2240  011120  004537  004106        LTST10:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 10
2241  011124  000020                        LINE10
2242          000011                        Y=Y+1
2243                                ;*************************************************************************
2244  011126  000037                RT37:   37                      ;ROUTINE # 37                           *
2245  011130  011144                        RT40                    ;ADDRESS OF NEXT TEST.                  *
2246  011132  000144                        100.                    ;ITERATION COUNT                        *
2247  011134  011136                        LTST11                  ;SCOPE ENTRY POINT                      *
2248          000037                        X=X+1
2249                                ;*************************************************************************
2250                                ;TRANSMITTER LINE TEST LINE 11
2251  011136  004537  004106        LTST11:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 11
2252  011142  000022                        LINE11
2253          000012                        Y=Y+1
2254                                ;*************************************************************************
2255  011144  000040                RT40:   40                      ;ROUTINE # 40                           *
2256  011146  011162                        RT41                    ;ADDRESS OF NEXT TEST.                  *
2257  011150  000144                        100.                    ;ITERATION COUNT                        *
2258  011152  011154                        LTST12                  ;SCOPE ENTRY POINT                      *
```

```
                000040                          X=X+1
                                        ;*******************************************************************************
                                        ;TRANSMITTER LINE TEST LINE 12
011154  004537  004106                  LTST12:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 12
011160  000024                                  LINE12
                000013                          Y=Y+1
                                        ;*******************************************************************************
011162  000041                          RT41:   41                      ;ROUTINE # 41                                  *
011164  011200                                  RT42                    ;ADDRESS OF NEXT TEST.                         *
011166  000144                                  100.                    ;ITERATION COUNT                               *
011170  011172                                  LTST13                  ;SCOPE ENTRY POINT                             *
                000041                          X=X+1
                                        ;*******************************************************************************
                                        ;TRANSMITTER LINE TEST LINE 13
011172  004537  004106                  LTST13:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 13
011176  000026                                  LINE13
                000014                          Y=Y+1
                                        ;*******************************************************************************
011200  000042                          RT42:   42                      ;ROUTINE # 42                                  *
011202  011216                                  RT43                    ;ADDRESS OF NEXT TEST.                         *
011204  000144                                  100.                    ;ITERATION COUNT                               *
011206  011210                                  LTST14                  ;SCOPE ENTRY POINT                             *
                000042                          X=X+1
                                        ;*******************************************************************************
                                        ;TRANSMITTER LINE TEST LINE 14
011210  004537  004106                  LTST14:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 14
011214  000030                                  LINE14
                000015                          Y=Y+1
                                        ;*******************************************************************************
011216  000043                          RT43:   43                      ;ROUTINE # 43                                  *
011220  011234                                  RT44                    ;ADDRESS OF NEXT TEST.                         *
011222  000144                                  100.                    ;ITERATION COUNT                               *
011224  011226                                  LTST15                  ;SCOPE ENTRY POINT                             *
                000043                          X=X+1
                                        ;*******************************************************************************
                                        ;TRANSMITTER LINE TEST LINE 15
011226  004537  004106                  LTST15:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 15
011232  000032                                  LINE15
                000016                          Y=Y+1
                                        ;*******************************************************************************
011234  000044                          RT44:   44                      ;ROUTINE # 44                                  *
011236  011252                                  RT45                    ;ADDRESS OF NEXT TEST.                         *
011240  000144                                  100.                    ;ITERATION COUNT                               *
011242  011244                                  LTST16                  ;SCOPE ENTRY POINT                             *
                000044                          X=X+1
                                        ;*******************************************************************************
                                        ;TRANSMITTER LINE TEST LINE 16
011244  004537  004106                  LTST16:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 16
011250  000034                                  LINE16
                000017                          Y=Y+1
                                        ;*******************************************************************************
011252  000045                          RT45:   45                      ;ROUTINE # 45                                  *
011254  011270                                  RT46                    ;ADDRESS OF NEXT TEST.                         *
011256  000144                                  100.                    ;ITERATION COUNT                               *
011260  011262                                  LTST17                  ;SCOPE ENTRY POINT                             *
                000045                          X=X+1
```

```
2315                                    ;*****************************************************************************
2316                                    ;TRANSMITTER LINE TEST LINE 17
2317  011262  004537  004106           LTST17:JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 17
2318  011266  000036                           LINE17
2319           000020                           Y=Y+1
2320           000000                           Y=0
2321           000000                           A=0
2322                                    ;*****************************************************************************
2323  011270  000046                   RT46:   46                      ;ROUTINE # 46                          *
2324  011272  011306                           RT47                    ;ADDRESS OF NEXT TEST                  *
2325  011274  000144                           100.                    ;ITERATION COUNT                       *
2326  011276  011300                           RCV0                    ;SCOPE ENTRY POINT                     *
2327           000046                           X=X+1
2328                                    ;*****************************************************************************
2329                                    ;RECEIVER LINE TEST LINE 0
2330  011300  004537  004320           RCV0:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 0
2331  011304  000000                           LINE0
2332           000001                           Y=Y+1
2333                                    ;*****************************************************************************
2334  011306  000047                   RT47:   47                      ;ROUTINE # 47                          *
2335  011310  011324                           RT50                    ;ADDRESS OF NEXT TEST                  *
2336  011312  000144                           100.                    ;ITERATION COUNT                       *
2337  011314  011316                           RCV1                    ;SCOPE ENTRY POINT                     *
2338           000047                           X=X+1
2339                                    ;*****************************************************************************
2340                                    ;RECEIVER LINE TEST LINE 1
2341  011316  004537  004320           RCV1:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 1
2342  011322  000002                           LINE1
2343           000002                           Y=Y+1
2344                                    ;*****************************************************************************
2345  011324  000050                   RT50:   50                      ;ROUTINE # 50                          *
2346  011326  011342                           RT51                    ;ADDRESS OF NEXT TEST                  *
2347  011330  000144                           100.                    ;ITERATION COUNT                       *
2348  011332  011334                           RCV2                    ;SCOPE ENTRY POINT                     *
2349           000050                           X=X+1
2350                                    ;*****************************************************************************
2351                                    ;RECEIVER LINE TEST LINE 2
2352  011334  004537  004320           RCV2:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 2
2353  011340  000004                           LINE2
2354           000003                           Y=Y+1
2355                                    ;*****************************************************************************
2356  011342  000051                   RT51:   51                      ;ROUTINE # 51                          *
2357  011344  011360                           RT52                    ;ADDRESS OF NEXT TEST                  *
2358  011346  000144                           100.                    ;ITERATION COUNT                       *
2359  011350  011352                           RCV3                    ;SCOPE ENTRY POINT                     *
2360           000051                           X=X+1
2361                                    ;*****************************************************************************
2362                                    ;RECEIVER LINE TEST LINE 3
2363  011352  004537  004320           RCV3:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 3
2364  011356  000006                           LINE3
2365           000004                           Y=Y+1
2366                                    ;*****************************************************************************
2367  011360  000052                   RT52:   52                      ;ROUTINE # 52                          *
2368  011362  011376                           RT53                    ;ADDRESS OF NEXT TEST                  *
2369  011364  000144                           100.                    ;ITERATION COUNT                       *
2370  011366  011370                           RCV4                    ;SCOPE ENTRY POINT                     *
```

```
2371            000052                          X=X+1
2372                    ;*****************************************************************************
2373                    ;RECEIVER LINE TEST LINE 4
2374  011370   004537  004320        RCV4:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 4
2375  011374   000010                        LINE4
2376            000005                        Y=Y+1
2377                    ;*****************************************************************************
2378  011376   000053                RT53:   53                      ;ROUTINE # 53                  *
2379  011400   011414                        RT54                    ;ADDRESS OF NEXT TEST          *
2380  011402   000144                        100.                    ;ITERATION COUNT               *
2381  011404   011406                        RCV5                    ;SCOPE ENTRY POINT             *
2382            000053                        X=X+1
2383                    ;*****************************************************************************
2384                    ;RECEIVER LINE TEST LINE 5
2385  011406   004537  004320        RCV5:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 5
2386  011412   000012                        LINE5
2387            000006                        Y=Y+1
2388                    ;*****************************************************************************
2389  011414   000054                RT54:   54                      ;ROUTINE # 54                  *
2390  011416   011432                        RT55                    ;ADDRESS OF NEXT TEST          *
2391  011420   000144                        100.                    ;ITERATION COUNT               *
2392  011422   011424                        RCV6                    ;SCOPE ENTRY POINT             *
2393            000054                        X=X+1
2394                    ;*****************************************************************************
2395                    ;RECEIVER LINE TEST LINE 6
2396  011424   004537  004320        RCV6:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 6
2397  011430   000014                        LINE6
2398            000007                        Y=Y+1
2399                    ;*****************************************************************************
2400  011432   000055                RT55:   55                      ;ROUTINE # 55                  *
2401  011434   011450                        RT56                    ;ADDRESS OF NEXT TEST          *
2402  011436   000144                        100.                    ;ITERATION COUNT               *
2403  011440   011442                        RCV7                    ;SCOPE ENTRY POINT             *
2404            000055                        X=X+1
2405                    ;*****************************************************************************
2406                    ;RECEIVER LINE TEST LINE 7
2407  011442   004537  004320        RCV7:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 7
2408  011446   000016                        LINE7
2409            000010                        Y=Y+1
2410                    ;*****************************************************************************
2411  011450   000056                RT56:   56                      ;ROUTINE # 56                  *
2412  011452   011466                        RT57                    ;ADDRESS OF NEXT TEST          *
2413  011454   000144                        100.                    ;ITERATION COUNT               *
2414  011456   011460                        RCV10                   ;SCOPE ENTRY POINT             *
2415            000056                        X=X+1
2416                    ;*****************************************************************************
2417                    ;RECEIVER LINE TEST LINE 10
2418  011460   004537  004320        RCV10:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 10
2419  011464   000020                        LINE10
2420            000011                        Y=Y+1
2421                    ;*****************************************************************************
2422  011466   000057                RT57:   57                      ;ROUTINE # 57                  *
2423  011470   011504                        RT60                    ;ADDRESS OF NEXT TEST          *
2424  011472   000144                        100.                    ;ITERATION COUNT               *
2425  011474   011476                        RCV11                   ;SCOPE ENTRY POINT             *
2426            000057                        X=X+1
```

```
2427                            ;*****************************************************************************
2428                            ;RECEIVER LINE TEST LINE 11
2429  011476  004537  004320   RCV11:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 11
2430  011502  000022                   LINE11
2431           000012                   Y=Y+1
2432                            ;*****************************************************************************
2433  011504  000060           RT60:   60                      ;ROUTINE # 60                              *
2434  011506  011522                   RT61                    ;ADDRESS OF NEXT TEST                      *
2435  011510  000144                   100.                    ;ITERATION COUNT                           *
2436  011512  011514                   RCV12                   ;SCOPE ENTRY POINT                         *
2437           000060                   X=X+1
2438                            ;*****************************************************************************
2439                            ;RECEIVER LINE TEST LINE 12
2440  011514  004537  004320   RCV12:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 12
2441  011520  000024                   LINE12
2442           000013                   Y=Y+1
2443                            ;*****************************************************************************
2444  011522  000061           RT61:   61                      ;ROUTINE # 61                              *
2445  011524  011540                   RT62                    ;ADDRESS OF NEXT TEST                      *
2446  011526  000144                   100.                    ;ITERATION COUNT                           *
2447  011530  011532                   RCV13                   ;SCOPE ENTRY POINT                         *
2448           000061                   X=X+1
2449                            ;*****************************************************************************
2450                            ;RECEIVER LINE TEST LINE 13
2451  011532  004537  004320   RCV13:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 13
2452  011536  000026                   LINE13
2453           000014                   Y=Y+1
2454                            ;*****************************************************************************
2455  011540  000062           RT62:   62                      ;ROUTINE # 62                              *
2456  011542  011556                   RT63                    ;ADDRESS OF NEXT TEST                      *
2457  011544  000144                   100.                    ;ITERATION COUNT                           *
2458  011546  011550                   RCV14                   ;SCOPE ENTRY POINT                         *
2459           000062                   X=X+1
2460                            ;*****************************************************************************
2461                            ;RECEIVER LINE TEST LINE 14
2462  011550  004537  004320   RCV14:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 14
2463  011554  000030                   LINE14
2464           000015                   Y=Y+1
2465                            ;*****************************************************************************
2466  011556  000063           RT63:   63                      ;ROUTINE # 63                              *
2467  011560  011574                   RT64                    ;ADDRESS OF NEXT TEST                      *
2468  011562  000144                   100.                    ;ITERATION COUNT                           *
2469  011564  011566                   RCV15                   ;SCOPE ENTRY POINT                         *
2470           000063                   X=X+1
2471                            ;*****************************************************************************
2472                            ;RECEIVER LINE TEST LINE 15
2473  011566  004537  004320   RCV15:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 15
2474  011572  000032                   LINE15
2475           000016                   Y=Y+1
2476                            ;*****************************************************************************
2477  011574  000064           RT64:   64                      ;ROUTINE # 64                              *
2478  011576  011612                   RT65                    ;ADDRESS OF NEXT TEST                      *
2479  011600  000144                   100.                    ;ITERATION COUNT                           *
2480  011602  011604                   RCV16                   ;SCOPE ENTRY POINT                         *
2481           000064                   X=X+1
2482                            ;*****************************************************************************
```

```
                                        ;RECEIVER LINE TEST LINE 16
011604  004537  004320          RCV16:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 16
011610  000034                          LINE16
        000017                          Y=Y+1
                                ;*****************************************************************************
011612  000065                  RT65:   65                      ;ROUTINE # 65                              *
011614  011632                          RT66                    ;ADDRESS OF NEXT TEST                      *
011616  000144                          100.                    ;ITERATION COUNT                           *
011620  011622                          RCV17                   ;SCOPE ENTRY POINT                         *
        000065                          X=X+1
                                ;*****************************************************************************
                                ;RECEIVER LINE TEST LINE 17
011622  004537  004320          RCV17:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 17
011626  000036                          LINE17
        000020                          Y=Y+1
011630  000240                          NOP
                                ;*****************************************************************************
011632  000066                  RT66:   66                      ;ROUTINE # 66                              *
011634  012072                          RT67                    ;ADDR OF NEXT ROUTINE.                     *
011636  000012                          10.                     ;ITERATION COUNT                           *
011640  011646                          RT66A                   ;SCOPE ENTRY POINT.                        *
        000066                          X=X+1
                                ;*****************************************************************************

011642  000240                          NOP
011644  000240                          NOP
                                ;TEST THAT NEX BIT (CSR BIT 14) SETS WHEN THE TRANSMITTER REFERENCES
                                ;NON-EXISTANT MEMORY, THE CORRESPONDING BAR BIT CLEARS
                                ;AND THAT AN INTERRUPT OCCURS. ALL LINES ARE USED FOR THE TEST.
011646  004737  003472          RT66A:  JSR     7,TIMER         ;GO CALCULATE MACINE TIME TO TRANSMIT
                                                                ;ONE CHARACTER
011652  012701  001106                  MOV     #CAT,%1         ;GET CAT ADDRESS
011656  012702  160000                  MOV     #160000,%2      ;GET A NON-EXISTANT ADDRESS
011662  012703  000020                  MOV     #16.,%3         ;GET COUNTER
011666  010221                  1$:     MOV     %2,(1)+         ;LOAD THE CURRENT ADDRESS
011670  005303                          DEC     %3              ;TABLE WITH NON-EXISTANT
011672  001375                          BNE     1$              ;ADDRESSES
011674  012701  000001                  MOV     #LBIT0,%1       ;GET LINE BIT
011700  012777  012036  167356          MOV     #6$,@XMTINT     ;LOAD TRANSMITTER INT. VECTOR
011706  052777  000060  167334  2$:     BIS     #60,@CSR        ;SET EXTENDED ADDRESS BITS
011714  050177  167332                  BIS     %1,@BAR         ;START TRANSMITTER
011720  013737  003644  011730          MOV     TIME14,3$       ;LOAD DELAY TIME TO
011726  104400                          DELAY                   ;DELAY FOR 1/4TH OF A CHARACTER
011730  000000                  3$:     OPEN                    ;TO RESPOND TO NEX
011732  017737  167314  001300          MOV     @BAR,RCVDAT     ;GET BAR DATA & TEST
011740  001406                          BEQ     4$              ;THAT IT IS CLEAR
011742  005037  001302                  CLR     XMTDAT
011746  104011                          ERROR1                  ;ERROR!BAR BIT DID NOT CLEAR
011750  005077  167276                  CLR     @BAR
011754  000440                          BR      7$              ;GO TO SCOPE
011756  032777  040000  167264  4$:     BIT     #BIT14,@CSR     ;TEST THAT NEX BIT IS SET
011764  001002                          BNE     5$              ;BRANCH IF SET
011766  104001                          ERROR                   ;ERROR! NEX BIT FAILED TO SET
011770  000432                          BR      7$              ;GO TO SCOPE
011772  042777  100000  167250  5$:     BIC     #BIT15,@CSR     ;CLEAR TRANSMITTER READY FLAG
012000  052777  010000  167242          BIS     #BIT12,@CSR     ;SET TRANSMITTER IE BIT
```

```
2539  012006  005037  177776          CLR     @#PSW           ;ALLOW INTERRUPTS
2540  012012  000240                  NOP
2541  012014  012737  000340  177776  MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
2542  012022  010137  001302          MOV     %1,XMTDAT       ;LOAD LINE THAT FAILED
2543  012026  005037  001300          CLR     RCVDAT
2544  012032  104011                  ERROR1                  ;ERROR! NEX FAILED TO CAUSE INTERRUPT
2545                                                          ;TYPEOUT SHOWS LINE # THAT FAILED
2546  012034  000410                  BR      7$              ;GO TO SCOPE
2547  012036  005077  167206  6$:     CLR     @CSR
2548  012042  012737  000340  177776  MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
2549  012050  022626                  CMP     (6)+,(6)+       ;ADJUST STACK PTR
2550  012052  006301                  ASL     %1              ;SHIFT LINE BIT
2551  012054  103314                  BCC     2$              ;DO NEXT LINE
2552  012056  013737  003642  012066  7$:     MOV     TIME1,8$        ;WAIT FOR TRANSMITTER TO RUN
2553  012064  104400                  DELAY                   ;TO COMPLETION BEFORE
2554  012066  000000          8$:     OPEN                    ;EXITING TEST
2555  012070  104006                  SCOPE                   ;SCOPE
2556                          ;*****************************************************************************
2557  012072  000067          RT67:   67                      ;ROUTINE # 67                               *
2558  012074  012202                  RT70                    ;ADDR OF NEXT ROUTINE.                      *
2559  012076  000012                  10.                     ;ITERATION COUNT                            *
2560  012100  012102                  RT67A                   ;SCOPE ENTRY POINT.                         *
2561          000067                  X=X+1
2562                          ;*****************************************************************************
2563
2564                          ;TEST THAT NEX BIT SETS IF THE DM11 TABLES ARE IN NON-EXISTANT CORE
2565  012102  012777  160000  167146  RT67A:  MOV     #160000,@BASREG ;SET BASE REGISTER TO NON=EXISTANT ADRS.
2566  012110  012737  012172  000004          MOV     #4$,@#ERRVEC    ;SET TIME OUT TRAP VECTOR
2567  012116  005737  160000                  TST     @#160000        ;CHECK THAT ADDRESS TIMES OUT
2568  012122  013737  003644  012140          MOV     TIME14,1$       ;GET TIME TO TRANSMIT 1/4 CHAR.
2569  012130  052777  000001  167114          BIS     #LBIT0,@BAR     ;START TO TRANSMIT ON LINE 0
2570  012136  104400                          DELAY                   ;DELAY 1/4TH OF A CHARACTER
2571  012140  000000          1$:     OPEN                    ;TIME
2572  012142  005077  167104                  CLR     @BAR            ;STOP TRANSMITTER
2573  012146  022777  040060  167074          CMP     #BIT14+60,@CSR  ;TEST THAT ONLY NEX IS SET
2574  012154  001401                          BEQ     2$
2575  012156  104001                          ERROR                   ;ERROR! EITHER NEX FALED TO SET
2576                                                                  ;OR OTHER BITS SET
2577  012160  013737  003642  012170  2$:     MOV     TIME1,3$        ;DELAY 1 CHARACTER TIME TO ALLOW
2578  012166  104400                          DELAY                   ;TRANSMITTER TO RUN TO
2579  012170  000000          3$:     OPEN                    ;COMPLETION
2580  012172  012737  000006  000004  4$:     MOV     #ERRVEC+2,@#ERRVEC      ;RESTORE TIME OUT TRAP
2581  012200  104006                          SCOPE
2582                          ;*****************************************************************************
2583  012202  000070          RT70:   70                      ;ROUTINE # 70                               *
2584  012204  012316                  RT71                    ;ADDR OF NEXT ROUTINE.                      *
2585  012206  000144                  100.                    ;ITERATION COUNT                            *
2586  012210  012212                  RT70A                   ;SCOPE ENTRY POINT.                         *
2587          000070                  X=X+1
2588                          ;*****************************************************************************
2589
2590                          ;TEST THAT WHEN THE GO BIT IS CLEAR THAT THE RECEIVERS DO NOT RECEIVE
2591                          ;DATA.EACH LINE IN TURN IS TRANSMITTED ON,AND WHEN TEN CHARACTERS
2592                          ;HAVE BEEN TRANSMITTED THE RECEIVER DONE FLAG IS TESTED. IF IT IS SET
2593                          ;AN ERROR IS INDICATED ON THE LINE DATA WAS RECEIVED ON.
2594  012212  005037  016176  RT70A:  CLR     LINE            ;SET UP TO TRANSMIT
```

```
2595  012216  004537  005532          1$:     JSR     5,@#XMITD       ;10. CHARACTERS
2596  012222  177766                          -10.                    ;ON EACH LINE
2597  012224  005777  167020          [illegible]  TST     @CSR            ;WAIT FOR 10. CHARACTERS
2598  012230  100375                          BPL     .-4             ;TO BE TRANSMITTED
2599  012232  042777  100000  167010          BIC     #100000,@CSR
2600  012240  105777  167004                  TSTB    @CSR            ;TEST RECEIVER DONE FLAG
2601  012244  100010                          BPL     2$
2602  012246  013737  001272  001300          MOV     LINBIT,RCVDAT   ;GET LINE BIT OF ACTIVE LINE
2603  012254  013737  001272  001302          MOV     LINBIT,XMTDAT   ;THAT ERROR OCCURED ON
2604  012262  104011                          ERROR1                  ;ERROR! DATA WAS RECEIVED ON LINE INDICATED
2605  012264  000413                          BR      4$              ;GO TO SCOPE
2606  012266  062737  000002  016176  2$:     ADD     #2,LINE         ;SET UP NEXT LINE NUMBER
2607  012274  006337  001272                  ASL     LINBIT          ;GET READY TO TRANSMIT ON NEXT LINE
2608  012300  103346                          BCC     1$              ;GO TRANSMIT ON NEXT LINE
2609  012302  013737  003642  012312          MOV     TIME1,3$
2610  012310  104400                          DELAY                   ;DELAY 1 CHARACTER
2611  012312  000000                  3$:     0                       ;TIME BEFORE ENTERING NEXT TEST
2612  012314  104006                  4$:     SCOPE                   ;SCOPE
2613                                  ;*************************************************************************
2614  012316  000071                  RT71:   71                      ;ROUTINE # 71                          *
2615  012320  012512                          RT72                    ;ADDR OF NEXT ROUTINE.                 *
2616  012322  000024                          20.                     ;ITERATION COUNT                       *
2617  012324  012326                          RT71A                   ;SCOPE ENTRY POINT.                    *
2618          000071                          X=X+1
2619                                  ;*************************************************************************
2620
2621                                  ;TEST THAT CURRENT ADDRESS INCREMENTS PROPERLY WHEN A CHAR-
2622                                  ;ACTER IS TRANSMITTED. LINE 0 IS USED FOR THE TEST.
2623  012326  005000                  RT71A:  CLR     %0              ;R0=CURRENT ADRS AFTER TRANSMISSION
2624  012330  010001                  1$:     MOV     %0,%1           ;R0=CURRENT ADDRESS BEFORE TRANSMISSION
2625  012332  005201                          INC     %1              ;AND R1=CURRENT ADDRESS AFTER TRANSMISSION
2626  012334  012737  012476  000004          MOV     #3$,@#ERRVEC            ;SET UP PROCESSOR
2627  012342  012737  000340  000006          MOV     #PRTY7,@#ERRVEC+2       ;TIME OUT TRAP
2628  012350  012777  001106  166700          MOV     #CAT,@BASREG    ;SET UP BASE REGISTER
2629  012356  010037  001106                  MOV     %0,CAT          ;LOAD CURRENT ADDRESS TABLE (LINE 0)
2630  012362  105710                          TSTB    (0)             ;DOES MEMORY EXIST?
2631  012364  012737  177776  001146          MOV     #-2,WCT         ;SET CHAR. COUNT TO TRANSMIT 1 CHAR.
2632  012372  012777  000005  166650          MOV     #5,@CSR         ;SET MAINT & GO BITS
2633  012400  012777  000001  166644          MOV     #LBIT0,@BAR     ;TRANSMIT ON LINE 0
2634  012406  105777  166636                  TSTB    @CSR            ;WAIT FOR THE RECEIVER
2635  012412  100375                          BPL     .-4             ;TO RECEIVE FIRST CHARACTER
2636  012414  042777  000200  166626          BIC     #200,@CSR       ;CLEAR RECEIVER DONE FLAG
2637  012422  105777  166622                  TSTB    @CSR            ;WAIT FOR RECEIVER TO RECEIVE
2638  012426  100375                          BPL     .-4             ;THE SECOND CHARACTER
2639  012430  023701  001106                  CMP     CAT,%1          ;TEST THAT CURRENT ADRS
2640                                                                  ;INCREMENTED PROPERLY
2641  012434  001413                          BEQ     2$
2642  012436  010137  001302                  MOV     %1,XMTDAT       ;GET COMPUTED RESULT
2643  012442  013737  001106  001300          MOV     CAT,RCVDAT      ;GET ACTUAL RESULT
2644  012450  104011                          ERROR1                  ;ERROR! CURRENT ADDRESS DID NOT
2645  012452  032777  040000  166422          BIT     #BIT14,@SWR     ;INCREMENT PROPERLY
2646  012460  001323                          BNE     1$              ;BRANCH IF SCOPE SWITCH IS SET
2647  012462  000405                          BR      3$              ;GO TO EXIT
2648  012464  005701                  2$:     TST     %1
2649  012466  001403                          BEQ     3$
2650  012470  000261                          SEC
```

```
                                        ROL     %0
                                        BMI     1$
000006  000004  3$:                   MOV     #ERRVEC+2,@#ERRVEC      ;RESTORE TIME OUT TRAP
000006                                CLR     @#ERRVEC+2
                                      SCOPE                           ;SCOPE
                                ;*****************************************************************************
                                RT72:   72                            ;ROUTINE # 72                          *
                                        RT73                          ;ADDR OF NEXT ROUTINE.                 *
                                        20.                           ;ITERATION COUNT                       *
                                        RT72A                         ;SCOPE ENTRY POINT.                    *
                                        X=X+1
                                ;*****************************************************************************

                                ;TEST THAT DATA CAN BE TRANSMITTED FROM ALL AVAILABLE CORE.
                                ;LINE 0 IS USED FOR THE TEST AND ONLY ONE WORD IS TRANSMITTED
                                ;AT A TIME.
                                RT72A:  CLR     %0                    ;CLEAR INDEX REGISTER
                                        RESET
013014  001106  1$:             MOV     AREA(0),CAT                   ;LOAD CURRENT ADDRESS
000005  166506                  MOV     #5,@CSR                       ;SET MAINT & GO BITS
177777  001146                  MOV     #-1,WCT                       ;SET UP CHAR COUNT TO TRANSMIT 1 CHAR.
000001  166474                  MOV     #LBIT0,@BAR                   ;TRANSMIT CHAR ON LINE 0
166466                          TST     @CSR                          ;WAIT FOR THE TRANSMITTER
                                BPL     .-4                           ;TO TRANSMIT THE CHARACTER
166460                          TSTB    @CSR                          ;TEST FOR DONE
                                BPL     .-4
166452                          CLR     @CSR                          ;CLEAR ALL FLAGS
001300                          CLR     RCVDAT
001306  001300                  MOVB    TUMTAB,RCVDAT                 ;GET RECEIVED CHARACTER
013014  001302                  MOVB    @AREA(0),XMTDAT               ;GET TRANSMITTED CHARACTER
001304  001302                  BIC     CARMSK,XMTDAT                 ;CLEAR NON-TRANSMITTED BITS
001300  001302                  CMPB    RCVDAT,XMTDAT                 ;COMPARE CHARACTERS
                                BEQ     2$                            ;BRANCH IF VALID COMPARISON
                                ERROR1                                ;ERROR!DATA COMPARISON ERROR
                        ;(CAT)-1 IS THE MEMORY LOCATION WHERE THE DATA WAS TRANSMITTED FROM
                                BR      6$                            ;GO TO EXIT
000006          2$:             CMP     %0,#6                         ;HAS FIRST 4K BEEN TESTED
                                BEQ     3$                            ;BRANCH IF IT HAS
                                TST     (0)+                          ;INCREMENT INDEX
                                BR      1$                            ;GO REPEAT TEST


000240          3$:             NOP                                   ;BEGIN TESTING ABOVE 4K
013002  000004                  MOV     #5$,@#ERRVEC                  ;SET TIME OUT TRAP TO EXIT
                                                                      ;TEST IF MEMORY TIMES OUT
                                CLR     %1                            ;SET UP DATA IDENTIFIER
                        4$:     INC     %1                            ;INCREMENT DATA IDENTIFIER
                                TST     (0)+                          ;INCREMENT INDEX
013014                          MOVB    %1,@AREA(0)                   ;LOAD IDENTIFIER INTO MEMORY
013014  001106                  MOV     AREA(0),CAT                   ;LOAD CURRENT ADDRESS
000005  166340                  MOV     #5,@CSR                       ;SET MAINT & GO BITS
177777  001146                  MOV     #-1,WCT                       ;SET UP CHAR COUNT TO TRANSMIT 1 CHAR.
000001  166326                  MOV     #LBIT0,@BAR                   ;TRANSMIT ON LINE 0
166320                          TST     @CSR                          ;WAIT FOR THE TRANSMITTER TO
                                BPL     .-4                           ;TRANSMIT THE CHARACTER
166312                          TSTB    @CSR                          ;TEST FOR CHARACTER DONE
```

```
2707  012736  100375                        BPL     .-4
2708  012740  005077  166304                CLR     @CSR
2709  012744  113737  001306  001300        MOVB    TUMTAB,RCVDAT   ;GET THE RECEIVED CHARACTER
2710  012752  117037  013014  001302        MOVB    @AREA(0),XMTDAT ;GET THE TRANSMITTED CHARACTER
2711  012760  043737  001304  001302        BIC     CARMSK,XMTDAT   ;CLEAR NON-TRANSMITTED BITS
2712  012766  123737  001300  001302        CMPB    RCVDAT,XMTDAT   ;COMPARE CHARACTERS
2713  012774  001733                        BEQ     4$              ;BRANCH IF VALID COMPARISON
2714  012776  104011                        ERROR1                  ;ERROR!DATA COMPARISON ERROR NUMBER
2715  013000  000404                        BR      6$              ;IN S/B GIVES MEMORY LOCATION (SEE TABLE
2716  013002  022626              5$:       POPSP2                  ;RESET THE STACK
2717  013004  012737  000006  000004        MOV     #6,@#ERRVEC     ;RESTORE TIME OUT TRAP
2718  013012  104006              6$:       SCOPE                   ;EXIT TEST
2719                                        ;MEMORY LOCATIONS TRANSMITTED FROM TABLE
2720  013014  000000              AREA:     0                       ;FOR DATA IN FIRST
2721  013016  005252                        5252                    ;4K SEE THE LISTING
2722  013020  012525                        12525                   ;CONTENTS OF THESE LOCATIONS (BYTE)
2723  013022  017777                        17777                   ;IS THE DATA TRANSMITTED
2724  013024  020000              A8K:      20000                   ;CONTENTS =1 (IF AVAILABLE)
2725  013026  026314                        26314                   ;    "     2       "
2726  013030  031463                        31463                   ;    "     3       "
2727
2728  013032  037477                        37477                   ;    "     4       "
2729  013034  040000              A12K:     40000                   ;    "     5       "
2730  013036  057477                        57477                   ;    "     6       "
2731  013040  060000              A16K:     60000                   ;    "     7       "
2732  013042  077477                        77477                   ;    "     10      "
2733  013044  100000              A20K:     100000                  ;    "     11      "
2734  013046  117477                        117477                  ;    "     12      "
2735  013050  120000              A24K:     120000                  ;    "     13      "
2736  013052  137477                        137477                  ;    "     14      "
2737  013054  140000              A28K:     140000                  ;    "     15      "
2738  013056  173000                        173000                  ;    "     16      "
2739                                        ;*************************************************************************
2740  013060  000073              RT73:     73                      ;ROUTINE # 73                                  *
2741  013062  013274                        RT74                    ;ADDR OF NEXT ROUTINE.                         *
2742  013064  000012                        10.                     ;ITERATION COUNT                               *
2743  013066  013070                        RT73A                   ;SCOPE ENTRY POINT.                            *
2744          000073                        X=X+1
2745                                        ;*************************************************************************
2746
2747                                        ;TEST THAT THE TRANSMITTER CAN TRANSMIT 100. CHARACTERS BEFORE SETTING
2748                                        ;THE READY BIT (CSR 15),AND CLEARING THE BAR BIT.
2749  013070  005037  016176              RT73A:  CLR     LINE
2750  013074  012777  000001  166146      1$:     MOV     #1,@CSR         ;SET THE GO BIT
2751  013102  005037  001300                      CLR     RCVDAT
2752  013106  013737  016176  001302              MOV     @#LINE,XMTDAT   ;GET LINE NUMBER (X2)
2753  013114  004537  005532                      JSR     5,@#XMITD       ;TRANSMIT 100. CHARACTERS
2754  013120  177634                              -100.                   ;ON LINE AS SPECIFIED BY LINE
2755  013122  105777  166122              2$:     TSTB    @CSR            ;WAIT FOR THE RECEIVER
2756  013126  100375                              BPL     2$              ;TO RECEIVE ONE CHARACTER
2757  013130  042777  000200  166112              BIC     #BIT7,@CSR      ;CLEAR CHAR. DONE FLAG
2758  013136  005237  001300                      INC     RCVDAT          ;INCREMENT CHAR. RCVD COUNT
2759  013142  023727  001300  000144              CMP     RCVDAT,#100.    ;HAVE 100. CHARS. BEEN RCVD
2760  013150  001416                              BEQ     4$
2761  013152  005777  166072                      TST     @CSR            ;TEST READY FLAG
2762  013156  100002                              BPL     3$              ;GO TEST BAR
```

```
2763  013160  104011                          ERROR1                  ;ERROR!READY BIT SET TOO SOON
2764                                  ;TYPEOUT SHOWS HOW MANY CHARS WERE RECEIVED WHEN READY SET AND THE LINE # (X2)
2765  013162  000443                          BR      8$              ;GO TO EXIT
2766
2767  013164  023777  001272  166060  3$:     CMP     @#LINBIT,@BAR   ;TEST THAT BAR BIT IS SET
2768  013172  001753                          BEQ     2$              ;BRANCH IF SET
2769  013174  017737  166052  001302          MOV     @BAR,XMTDAT     ;GET BAR CONTENTS
2770  013202  104011                          ERROR1                  ;ERROR! BAR BIT CLEARED TO SOON
2771                                  ;TYPEOUT SHOWS THE BAR CONTENTS AND HOW MANY CHARS WERE RECEIVED WHEN BAR FAILED
2772                                  ;LOCATION LINBIT HAS THE CORRECT BAR CONTENTS.
2773  013204  000432                          BR      8$              ;EXIT TEST
2774  013206  013737  003644  013216  4$:     MOV     TIME14,5$       ;DELAY 1/4 CHARACTER TIME
2775  013214  104400                          DELAY                   ;TO ALLOW TRANSMITTER TO FINISH
2776  013216  000000                  5$:     OPEN
2777  013220  005777  166024                  TST     @CSR            ;TEST READY FLAG (SHOULD BE SET)
2778  013224  100402                          BMI     6$              ;GO TEST BAR
2779  013226  104001                          ERROR                   ;ERROR! READY FLAG FAILED TO SET
2780  013230  000420                          BR      8$              ;GO TO EXIT
2781  013232  005777  166014          6$:     TST     @BAR            ;TEST THAT BAR BIT IS CLEAR
2782  013236  001407                          BEQ     7$              ;GO TO 7$ IF CLEAR
2783  013240  017737  166006  001300          MOV     @BAR,RCVDAT
2784  013246  005037  001302                  CLR     XMTDAT
2785  013252  104011                          ERROR1                  ;ERROR! BAR BIT FAILED TO CLEAR
2786  013254  000406                          BR      8$
2787  013256  062737  000002  016176  7$:     ADD     #2,@#LINE
2788  013264  006337  001272                  ASL     @#LINBIT
2789  013270  103301                          BCC     1$
2790  013272  104006                  8$:     SCOPE
2791                                  ;*************************************************************************
2792  013274  000074                  RT74:   74                      ;ROUTINE # 74                          *
2793  013276  013462                          RT75                    ;ADDR OF NEXT ROUTINE.                 *
2794  013300  000012                          10.                     ;ITERATION COUNT                       *
2795  013302  013304                          RT74A                   ;SCOPE ENTRY POINT.                    *
2796          000074                          X=X+1
2797                                  ;*************************************************************************
2798
2799                                  ;TEST THAT THE TUMBLE TABLE POINTER INCREMENTS PROPERLY AND
2800                                  ;RETURNS TO THE BEGINNING AFTER 64. CHARACTERS HAVE BEEN RECEIVED
2801                                  ;LINE 0 IS USED FOR THE TEST.
2802  013304  012701  001306  RT74A:  MOV     #TUMTAB,%1      ;CLEAR THE
2803  013310  012702  000100          MOV     #64.,%2         ;TUMBLE TABLE
2804  013314  005021                  1$:     CLR     (1)+
2805  013316  005302                          DEC     %2
2806  013320  001375                          BNE     1$
2807  013322  012701  001306                  MOV     #TUMTAB,%1
2808  013326  012777  000004  165714          MOV     #BIT2,@CSR      ;SET MAINT BIT & CLEAR GO BIT
2809  013334  005037  001302                  CLR     XMTDAT
2810  013340  005037  001300                  CLR     RCVDAT
2811  013344  012737  177677  001146          MOV     #-65.,WCT       ;SET UP TO TRANSMIT 65. CHARACTERS
2812  013352  052777  000001  165670          BIS     #BIT0,@CSR      ;SET THE GO BIT
2813  013360  012777  000001  165664          MOV     #LBIT0,@BAR     ;TRANSMIT ON LINE 0
2814  013366  105777  165656          2$:     TSTB    @CSR            ;WAIT FOR CHAR. DONE FLAG
2815  013372  100375                          BPL     2$
2816  013374  042777  000200  165646          BIC     #BIT7,  @CSR    ;CLEAR CHAR. DONE FLAG
2817  013402  005237  001300                  INC     RCVDAT          ;INCREMENT CHARACTERS
2818  013406  005237  001302                  INC     XMTDAT          ;RECEIVED COUNT
```

```
 2819  013412  005711                          TST     (1)             ;TEST TT ENTRY FOR VALID
 2820  013414  100402                          BMI     3$              ;DATA ENTRY
 2821  013416  104011                          ERROR1                  ;ERROR! NO VALID DATA ENTRY
 2822                                                                  ;TYPEOUT SHOWS # OF CHARS. RCVD WHEN ERROR OCCURED
 2823  013420  000417                          BR      4$              ;GO TO SCOPE
 2824  013422  005021                  3$:     CLR     (1)+            ;CLEAR TT ENTRY
 2825  013424  023727  001302  000100         CMP     XMTDAT,#64.     ;HAVE 64. CHARACTERS BEEN RECEIVED
 2826  013432  001355                          BNE     2$
 2827  013434  005777  165610                  TST     @CSR            ;WAIT FOR THE LAST CHARACTER
 2828  013440  100375                          BPL     .-4             ;TO BE TRANSMITTED
 2829  013442  105777  165602                  TSTB    @CSR            ;TEST FOR DONE
 2830  013446  100375                          BPL     .-4
 2831  013450  005737  001306                  TST     TUMTAB          ;TEST FIRST TT ENTRY
 2832  013454  100401                          BMI     4$              ;FOR VALID DATA
 2833  013456  104001                          ERROR                   ;ERROR! POINTER DID NOT RETURN
 2834  013460  104006                  4$:     SCOPE                   ;SCOPE
 2835          000000                          A=0
 2836          000000                          Y=0
 2837                          ;*****************************************************************************
 2838  013462  000075                  RT75:   75                      ;ROUTINE # 75                        *
 2839  013464  013500                          RT76                    ;ADDRESS OF NEXT TEST                *
 2840  013466  000144                          100.                    ;ITERATION COUNT                     *
 2841  013470  013472                          BRK0                    ;SCOPE ENTRY POINT                   *
 2842          000075                          X=X+1
 2843                          ;*****************************************************************************
 2844                          ;BREAK TEST ON LINE 0.
 2845  013472  004537  004742          BRK0:   JSR     5,BRKTST        ;GO DO BREAK TEST
 2846  013476  000001                          LBIT0                   ;ON LINE 0
 2847          000001                          Y=Y+1
 2848                          ;*****************************************************************************
 2849  013500  000076                  RT76:   76                      ;ROUTINE # 76                        *
 2850  013502  013516                          RT77                    ;ADDRESS OF NEXT TEST                *
 2851  013504  000144                          100.                    ;ITERATION COUNT                     *
 2852  013506  013510                          BRK1                    ;SCOPE ENTRY POINT                   *
 2853          000076                          X=X+1
 2854                          ;*****************************************************************************
 2855                          ;BREAK TEST ON LINE 1.
 2856  013510  004537  004742          BRK1:   JSR     5,BRKTST        ;GO DO BREAK TEST
 2857  013514  000002                          LBIT1                   ;ON LINE 1
 2858          000002                          Y=Y+1
 2859                          ;*****************************************************************************
 2860  013516  000077                  RT77:   77                      ;ROUTINE # 77                        *
 2861  013520  013534                          RT100                   ;ADDRESS OF NEXT TEST                *
 2862  013522  000144                          100.                    ;ITERATION COUNT                     *
 2863  013524  013526                          BRK2                    ;SCOPE ENTRY POINT                   *
 2864          000077                          X=X+1
 2865                          ;*****************************************************************************
 2866                          ;BREAK TEST ON LINE 2.
 2867  013526  004537  004742          BRK2:   JSR     5,BRKTST        ;GO DO BREAK TEST
 2868  013532  000004                          LBIT2                   ;ON LINE 2
 2869          000003                          Y=Y+1
 2870                          ;*****************************************************************************
 2871  013534  000100                  RT100:  100                     ;ROUTINE # 100                       *
 2872  013536  013552                          RT101                   ;ADDRESS OF NEXT TEST                *
 2873  013540  000144                          100.                    ;ITERATION COUNT                     *
 2874  013542  013544                          BRK3                    ;SCOPE ENTRY POINT                   *
```

```
                000100                          X=X+1
                                        ;*****************************************************************************
                                        ;BREAK TEST ON LINE 3.
013544  004537  004742          BRK3:   JSR     5,BRKTST        ;GO DO BREAK TEST
013550  000010                          LBIT3                   ;ON LINE 3
                000004                  Y=Y+1
                                        ;*****************************************************************************
013552  000101                  RT101:  101                     ;ROUTINE # 101                                  *
013554  013570                          RT102                   ;ADDRESS OF NEXT TEST                           *
013556  000144                          100.                    ;ITERATION COUNT                                *
013560  013562                          BRK4                    ;SCOPE ENTRY POINT                              *
                000101                  X=X+1
                                        ;*****************************************************************************
                                        ;BREAK TEST ON LINE 4.
013562  004537  004742          BRK4:   JSR     5,BRKTST        ;GO DO BREAK TEST
013566  000020                          LBIT4                   ;ON LINE 4
                000005                  Y=Y+1
                                        ;*****************************************************************************
013570  000102                  RT102:  102                     ;ROUTINE # 102                                  *
013572  013606                          RT103                   ;ADDRESS OF NEXT TEST                           *
013574  000144                          100.                    ;ITERATION COUNT                                *
013576  013600                          BRK5                    ;SCOPE ENTRY POINT                              *
                000102                  X=X+1
                                        ;*****************************************************************************
                                        ;BREAK TEST ON LINE 5.
013600  004537  004742          BRK5:   JSR     5,BRKTST        ;GO DO BREAK TEST
013604  000040                          LBIT5                   ;ON LINE 5
                000006                  Y=Y+1
                                        ;*****************************************************************************
013606  000103                  RT103:  103                     ;ROUTINE # 103                                  *
013610  013624                          RT104                   ;ADDRESS OF NEXT TEST                           *
013612  000144                          100.                    ;ITERATION COUNT                                *
013614  013616                          BRK6                    ;SCOPE ENTRY POINT                              *
                000103                  X=X+1
                                        ;*****************************************************************************
                                        ;BREAK TEST ON LINE 6.
013616  004537  004742          BRK6:   JSR     5,BRKTST        ;GO DO BREAK TEST
013622  000100                          LBIT6                   ;ON LINE 6
                000007                  Y=Y+1
                                        ;*****************************************************************************
013624  000104                  RT104:  104                     ;ROUTINE # 104                                  *
013626  013642                          RT105                   ;ADDRESS OF NEXT TEST                           *
013630  000144                          100.                    ;ITERATION COUNT                                *
013632  013634                          BRK7                    ;SCOPE ENTRY POINT                              *
                000104                  X=X+1
                                        ;*****************************************************************************
                                        ;BREAK TEST ON LINE 7.
013634  004537  004742          BRK7:   JSR     5,BRKTST        ;GO DO BREAK TEST
013640  000200                          LBIT7                   ;ON LINE 7
                000010                  Y=Y+1
                                        ;*****************************************************************************
013642  000105                  RT105:  105                     ;ROUTINE # 105                                  *
013644  013660                          RT106                   ;ADDRESS OF NEXT TEST                           *
013646  000144                          100.                    ;ITERATION COUNT                                *
013650  013652                          BRK10                   ;SCOPE ENTRY POINT                              *
                000105                  X=X+1
```

```
                                ;*******************************************************************************
                                ;BREAK TEST ON LINE 10.
013652  004537  004742          BRK10:  JSR     5,BRKTST        ;GO DO BREAK TEST
013656  000400                          LBIT10                  ;ON LINE 10
        000011                          Y=Y+1
                                ;*******************************************************************************
013660  000106                  RT106:  106                     ;ROUTINE # 106                                *
013662  013676                          RT107                   ;ADDRESS OF NEXT TEST                         *
013664  000144                          100.                    ;ITERATION COUNT                              *
013666  013670                          BRK11                   ;SCOPE ENTRY POINT                            *
        000106                          X=X+1
                                ;*******************************************************************************
                                ;BREAK TEST ON LINE 11.
013670  004537  004742          BRK11:  JSR     5,BRKTST        ;GO DO BREAK TEST
013674  001000                          LBIT11                  ;ON LINE 11
        000012                          Y=Y+1
                                ;*******************************************************************************
013676  000107                  RT107:  107                     ;ROUTINE # 107                                *
013700  013714                          RT110                   ;ADDRESS OF NEXT TEST                         *
013702  000144                          100.                    ;ITERATION COUNT                              *
013704  013706                          BRK12                   ;SCOPE ENTRY POINT                            *
        000107                          X=X+1
                                ;*******************************************************************************
                                ;BREAK TEST ON LINE 12.
013706  004537  004742          BRK12:  JSR     5,BRKTST        ;GO DO BREAK TEST
013712  002000                          LBIT12                  ;ON LINE 12
        000013                          Y=Y+1
                                ;*******************************************************************************
013714  000110                  RT110:  110                     ;ROUTINE # 110                                *
013716  013732                          RT111                   ;ADDRESS OF NEXT TEST                         *
013720  000144                          100.                    ;ITERATION COUNT                              *
013732  013724                          BRK13                   ;SCOPE ENTRY POINT                            *
        000110                          X=X+1
                                ;*******************************************************************************
                                ;BREAK TEST ON LINE 13.
013724  004537  004742          BRK13:  JSR     5,BRKTST        ;GO DO BREAK TEST
013730  004000                          LBIT13                  ;ON LINE 13
        000014                          Y=Y+1
                                ;*******************************************************************************
013732  000111                  RT111:  111                     ;ROUTINE # 111                                *
013734  013750                          RT112                   ;ADDRESS OF NEXT TEST                         *
013736  000144                          100.                    ;ITERATION COUNT                              *
013740  013742                          BRK14                   ;SCOPE ENTRY POINT                            *
        000111                          X=X+1
                                ;*******************************************************************************
                                ;BREAK TEST ON LINE 14.
013742  004537  004742          BRK14:  JSR     5,BRKTST        ;GO DO BREAK TEST
013746  010000                          LBIT14                  ;ON LINE 14
        000015                          Y=Y+1
                                ;*******************************************************************************
013750  000112                  RT112:  112                     ;ROUTINE # 112                                *
013752  013766                          RT113                   ;ADDRESS OF NEXT TEST                         *
013754  000144                          100.                    ;ITERATION COUNT                              *
013756  013760                          BRK15                   ;SCOPE ENTRY POINT                            *
        000112                          X=X+1
                                ;*******************************************************************************
```

```
                                ;BREAK TEST ON LINE 15.
013760  004537  004742  BRK15:  JSR     5,BRKTST        ;GO DO BREAK TEST
013764  020000                  LBIT15                  ;ON LINE 15
        000016                  Y=Y+1
                        ;*************************************************************************
013766  000113          RT113:  113                     ;ROUTINE # 113                          *
013770  014004                  RT114                   ;ADDRESS OF NEXT TEST                   *
013772  000144                  100.                    ;ITERATION COUNT                        *
013774  013776                  BRK16                   ;SCOPE ENTRY POINT                      *
        000113                  X=X+1
                        ;*************************************************************************
                        ;BREAK TEST ON LINE 16.
013776  004537  004742  BRK16:  JSR     5,BRKTST        ;GO DO BREAK TEST
014002  040000                  LBIT16                  ;ON LINE 16
        000017                  Y=Y+1
                        ;*************************************************************************
014004  000114          RT114:  114                     ;ROUTINE # 114                          *
014006  014022                  RT115                   ;ADDRESS OF NEXT TEST                   *
014010  000144                  100.                    ;ITERATION COUNT                        *
014012  014014                  BRK17                   ;SCOPE ENTRY POINT                      *
        000114                  X=X+1
                        ;*************************************************************************
                        ;BREAK TEST ON LINE 17.
014014  004537  004742  BRK17:  JSR     5,BRKTST        ;GO DO BREAK TEST
014020  100000                  LBIT17                  ;ON LINE 17
        000020                  Y=Y+1
        000000                  A=0
        000000                  Y=0
                        ;*************************************************************************
014022  000115          RT115:  115                     ;ROUTINE #115                           *
014024  014040                  RT116                   ;ADDRESS OF NEXT TEST                   *
014026  000144                  100.                    ;ITERATION COUNT                        *
014030  014032                  DAT0                    ;SCOPE ENTRY POINT                      *
        000115                  X=X+1
                        ;*************************************************************************
                        ;DATA TEST 100 CHARACTERS LINE0
014032  004537  005576  DAT0:   JSR     5,DATTST        ;GO RUN DATA TEST
014036  000000                  LINE0                   ;ON LINE0
        000001                  Y=Y+1
                        ;*************************************************************************
014040  000116          RT116:  116                     ;ROUTINE #116                           *
014042  014056                  RT117                   ;ADDRESS OF NEXT TEST                   *
014044  000144                  100.                    ;ITERATION COUNT                        *
014046  014050                  DAT1                    ;SCOPE ENTRY POINT                      *
        000116                  X=X+1
                        ;*************************************************************************
                        ;DATA TEST 100 CHARACTERS LINE1
014050  004537  005576  DAT1:   JSR     5,DATTST        ;GO RUN DATA TEST
014054  000002                  LINE1                   ;ON LINE1
        000002                  Y=Y+1
                        ;*************************************************************************
014056  000117          RT117:  117                     ;ROUTINE #117                           *
014060  014074                  RT120                   ;ADDRESS OF NEXT TEST                   *
014062  000144                  100.                    ;ITERATION COUNT                        *
014064  014066                  DAT2                    ;SCOPE ENTRY POINT                      *
        000117                  X=X+1
```

```
3043                                    ;*****************************************************************************
3044                                    ;DATA TEST 100 CHARACTERS LINE2
3045  014066  004537  005576            DAT2:   JSR     5,DATTST        ;GO RUN DATA TEST
3046  014072  000004                            LINE2                   ;ON LINE2
3047          000003                            Y=Y+1
3048                                    ;*****************************************************************************
3049  014074  000120                    RT120:  120                     ;ROUTINE #120                           *
3050  014076  014112                            RT121                   ;ADDRESS OF NEXT TEST                   *
3051  014100  000144                            100.                    ;ITERATION COUNT                        *
3052  014102  014104                            DAT3                    ;SCOPE ENTRY POINT                      *
3053          000120                            X=X+1
3054                                    ;*****************************************************************************
3055                                    ;DATA TEST 100 CHARACTERS LINE3
3056  014104  004537  005576            DAT3:   JSR     5,DATTST        ;GO RUN DATA TEST
3057  014110  000006                            LINE3                   ;ON LINE3
3058          000004                            Y=Y+1
3059                                    ;*****************************************************************************
3060  014112  000121                    RT121:  121                     ;ROUTINE #121                           *
3061  014114  014130                            RT122                   ;ADDRESS OF NEXT TEST                   *
3062  014116  000144                            100.                    ;ITERATION COUNT                        *
3063  014120  014122                            DAT4                    ;SCOPE ENTRY POINT                      *
3064          000121                            X=X+1
3065                                    ;*****************************************************************************
3066                                    ;DATA TEST 100 CHARACTERS LINE4
3067  014122  004537  005576            DAT4:   JSR     5,DATTST        ;GO RUN DATA TEST
3068  014126  000010                            LINE4                   ;ON LINE4
3069          000005                            Y=Y+1
3070                                    ;*****************************************************************************
3071  014130  000122                    RT122:  122                     ;ROUTINE #122                           *
3072  014132  014146                            RT123                   ;ADDRESS OF NEXT TEST                   *
3073  014134  000144                            100.                    ;ITERATION COUNT                        *
3074  014136  014140                            DAT5                    ;SCOPE ENTRY POINT                      *
3075          000122                            X=X+1
3076                                    ;*****************************************************************************
3077                                    ;DATA TEST 100 CHARACTERS LINE5
3078  014140  004537  005576            DAT5:   JSR     5,DATTST        ;GO RUN DATA TEST
3079  014144  000012                            LINE5                   ;ON LINE5
3080          000006                            Y=Y+1
3081                                    ;*****************************************************************************
3082  014146  000123                    RT123:  123                     ;ROUTINE #123                           *
3083  014150  014164                            RT124                   ;ADDRESS OF NEXT TEST                   *
3084  014152  000144                            100.                    ;ITERATION COUNT                        *
3085  014154  014156                            DAT6                    ;SCOPE ENTRY POINT                      *
3086          000123                            X=X+1
3087                                    ;*****************************************************************************
3088                                    ;DATA TEST 100 CHARACTERS LINE6
3089  014156  004537  005576            DAT6:   JSR     5,DATTST        ;GO RUN DATA TEST
3090  014162  000014                            LINE6                   ;ON LINE6
3091          000007                            Y=Y+1
3092                                    ;*****************************************************************************
3093  014164  000124                    RT124:  124                     ;ROUTINE #124                           *
3094  014166  014202                            RT125                   ;ADDRESS OF NEXT TEST                   *
3095  014170  000144                            100.                    ;ITERATION COUNT                        *
3096  014172  014174                            DAT7                    ;SCOPE ENTRY POINT                      *
3097          000124                            X=X+1
3098                                    ;*****************************************************************************
```

```
                                        ;DATA TEST 100 CHARACTERS LINE7
014174  004537  005576        DAT7:   JSR     5,DATTST        ;GO RUN DATA TEST
014200  000016                        LINE7                   ;ON LINE7
        000010                        Y=Y+1
                                ;*****************************************************************************
014202  000125                RT125:  125                     ;ROUTINE #125                                  *
014204  014220                        RT126                   ;ADDRESS OF NEXT TEST                          *
014206  000144                        100.                    ;ITERATION COUNT                               *
014210  014212                        DAT10                   ;SCOPE ENTRY POINT                             *
        000125                        X=X+1
                                ;*****************************************************************************
                                ;DATA TEST 100 CHARACTERS LINE10
014212  004537  005576        DAT10:  JSR     5,DATTST        ;GO RUN DATA TEST
014216  000020                        LINE10                  ;ON LINE10
        000011                        Y=Y+1
                                ;*****************************************************************************
014220  000126                RT126:  126                     ;ROUTINE #126                                  *
014222  014236                        RT127                   ;ADDRESS OF NEXT TEST                          *
014224  000144                        100.                    ;ITERATION COUNT                               *
014226  014230                        DAT11                   ;SCOPE ENTRY POINT                             *
        000126                        X=X+1
                                ;*****************************************************************************
                                ;DATA TEST 100 CHARACTERS LINE11
014230  004537  005576        DAT11:  JSR     5,DATTST        ;GO RUN DATA TEST
014234  000022                        LINE11                  ;ON LINE11
        000012                        Y=Y+1
                                ;*****************************************************************************
014236  000127                RT127:  127                     ;ROUTINE #127                                  *
014240  014254                        RT130                   ;ADDRESS OF NEXT TEST                          *
014242  000144                        100.                    ;ITERATION COUNT                               *
014244  014246                        DAT12                   ;SCOPE ENTRY POINT                             *
        000127                        X=X+1
                                ;*****************************************************************************
                                ;DATA TEST 100 CHARACTERS LINE12
014246  004537  005576        DAT12:  JSR     5,DATTST        ;GO RUN DATA TEST
014252  000024                        LINE12                  ;ON LINE12
        000013                        Y=Y+1
                                ;*****************************************************************************
014254  000130                RT130:  130                     ;ROUTINE #130                                  *
014256  014272                        RT131                   ;ADDRESS OF NEXT TEST                          *
014260  000144                        100.                    ;ITERATION COUNT                               *
014262  014264                        DAT13                   ;SCOPE ENTRY POINT                             *
        000130                        X=X+1
                                ;*****************************************************************************
                                ;DATA TEST 100 CHARACTERS LINE13
014264  004537  005576        DAT13:  JSR     5,DATTST        ;GO RUN DATA TEST
014270  000026                        LINE13                  ;ON LINE13
        000014                        Y=Y+1
                                ;*****************************************************************************
014272  000131                RT131:  131                     ;ROUTINE #131                                  *
014274  014310                        RT132                   ;ADDRESS OF NEXT TEST                          *
014276  000144                        100.                    ;ITERATION COUNT                               *
014300  014302                        DAT14                   ;SCOPE ENTRY POINT                             *
        000131                        X=X+1
                                ;*****************************************************************************
                                ;DATA TEST 100 CHARACTERS LINE14
```

```
3155  014302  004537  005576          DAT14:  JSR     5,DATTST        ;GO RUN DATA TEST
3156  014306  000030                          LINE14                  ;ON LINE14
3157          000015                          Y=Y+1
3158                                  ;*****************************************************************************
3159  014310  000132                  RT132:  132                     ;ROUTINE #132                             *
3160  014312  014326                          RT133                   ;ADDRESS OF NEXT TEST                     *
3161  014314  000144                          100.                    ;ITERATION COUNT                          *
3162  014316  014320                          DAT15                   ;SCOPE ENTRY POINT                        *
3163          000132                          X=X+1
3164                                  ;*****************************************************************************
3165                                  ;DATA TEST 100 CHARACTERS LINE15
3166  014320  004537  005576          DAT15:  JSR     5,DATTST        ;GO RUN DATA TEST
3167  014324  000032                          LINE15                  ;ON LINE15
3168          000016                          Y=Y+1
3169                                  ;*****************************************************************************
3170  014326  000133                  RT133:  133                     ;ROUTINE #133                             *
3171  014330  014344                          RT134                   ;ADDRESS OF NEXT TEST                     *
3172  014332  000144                          100.                    ;ITERATION COUNT                          *
3173  014334  014336                          DAT16                   ;SCOPE ENTRY POINT                        *
3174          000133                          X=X+1
3175                                  ;*****************************************************************************
3176                                  ;DATA TEST 100 CHARACTERS LINE16
3177  014336  004537  005576          DAT16:  JSR     5,DATTST        ;GO RUN DATA TEST
3178  014342  000034                          LINE16                  ;ON LINE16
3179          000017                          Y=Y+1
3180                                  ;*****************************************************************************
3181  014344  000134                  RT134:  134                     ;ROUTINE #134                             *
3182  014346  014362                          RT135                   ;ADDRESS OF NEXT TEST                     *
3183  014350  000144                          100.                    ;ITERATION COUNT                          *
3184  014352  014354                          DAT17                   ;SCOPE ENTRY POINT                        *
3185          000134                          X=X+1
3186                                  ;*****************************************************************************
3187                                  ;DATA TEST 100 CHARACTERS LINE17
3188  014354  004537  005576          DAT17:  JSR     5,DATTST        ;GO RUN DATA TEST
3189  014360  000036                          LINE17                  ;ON LINE17
3190          000020                          Y=Y+1
3191                                  ;*****************************************************************************
3192  014362  000135                  RT135:  135                     ;ROUTINE # 135                            *
3193  014364  014762                          RT136                   ;ADDR OF NEXT ROUTINE.                    *
3194  014366  000144                          100.                    ;ITERATION COUNT                          *
3195  014370  014372                          RT135A                  ;SCOPE ENTRY POINT.                       *
3196          000135                          X=X+1
3197                                  ;*****************************************************************************
3198
3199                                  ;TEST THAT DATA (ALL 1'S) CAN BE TRANSMITTED ON LINES SIMULTANEOUSLY.
3200                                  ;THE FOLLOWING TESTS ARE PERFORMED:
3201                                  ;       THERE ARE 16. DATA ENTRIES
3202                                  ;       THERE ISN'T A 17TH ENTRY
3203                                  ;       DATA RECEIVED IS CORRECT
3204                                  ;       ONE DATA ENTRY PER LINE
3205
3206  014372  005037  001306          RT135A: CLR     TUMTAB          ;CLEAR THE
3207  014376  004537  005510                  JSR     5,BMOVE         ;TUMBLE
3208  014402  001306                          TUMTAB                  ;TABLE
3209  014404  001307                          TUMTAB+1                ;(200
3210  014406  000177                          177                     ;ENTRIES)
```

```
3211  014410  012737  177777  016350        MOV     #-1,OUTBUF      ;LOAD CHAR INTO OUTPUT BUFFER
3212  014416  005000                        CLR     %0              ;SET R0 = LINE 0
3213  014420  012737  000001  001272        MOV     #LBIT0,LINBIT   ;GET LINE BIT
3214  014426  012777  000001  164614        MOV     #BIT0,@CSR      ;SET THE GO BIT
3215  014434  010037  016176          1$:   MOV     %0,LINE         ;GET LINE NUMBER
3216  014440  004537  005532                JSR     5,XMITD         ;TRANSMIT 1 CHAR.
3217  014444  177777                        -1                      ;ON EACH LINE
3218  014446  005720                        TST     (0)+            ;INCREMENT LINE NUMBER (+2)
3219  014450  006337  001272                ASL     LINBIT          ;SHIFT LINE BIT TO NEXT LINE
3220  014454  103367                        BCC     1$              ;BRANCH IF ALL LINES NOT DONE
3221  014456  013737  003642  014466        MOV     TIME1,2$        ;PUT TIME TO TRANSMIT 1 CHAR
3222  014464  104400                        DELAY                   ;DELAY 1
3223  014466  000000                  2$:   OPEN                    ;CHARACTER TIME
3224  014470  017737  164556  001300        MOV     @BAR,RCVDAT     ;GET & TEST BAR CONTENTS
3225  014476  001410                        BEQ     3$              ;BRANCH IF 0
3226  014500  005037  001302                CLR     XMTDAT
3227  014504  104011                        ERROR1                  ;ERROR! BAR NOT CLEAR AFTER ALL
3228  014506  005077  164540                CLR     @BAR            ;LINES FINISHED
3229  014512  005077  164532                CLR     @CSR
3230  014516  000520                        BR      16$             ;GO TO EXIT
3231  014520  032777  020000  164522  3$:   BIT     #BIT13,@CSR     ;TEST THAT OVER RUN DID NOT SET
3232  014526  001404                        BEQ     4$
3233  014530  104001                        ERROR                   ;ERROR! OVER RUN BIT SET
3234  014532  005077  164512                CLR     @CSR
3235  014536  000510                        BR      16$             ;GO TO EXIT
3236
3237                                  ;TEST THAT THERE ARE 16. VALID DATA ENTRIES
3238  014540  005077  164504          4$:   CLR     @^CSR           ;CLEAR THE CSR
3239  014544  012702  000020                MOV     #16.,%2         ;GET TT SCAN COUNT
3240  014550  012701  001306                MOV     #TUMTAB,%1      ;GET FIRST TT ADDRESS
3241  014554  005302                  5$:   DEC     %2              ;DECREMENT SCAN COUNTER
3242  014556  100404                        BMI     6$              ;BRANCH IF 16. ENTRIES SCANNED
3243  014560  005721                        TST     (1)+            ;TEST FOR VALID DATA ENTRY
3244  014562  100774                        BMI     5$              ;BRANCH IF FOUND
3245  014564  104001                        ERROR                   ;ERROR! MISING DATA ENTRY
3246  014566  000474                        BR      16$             ;GO TO EXIT
3247  014570  005721                  6$:   TST     (1)+            ;TEST 17TH ENTRY (SHOULD BE = TO 0)
3248  014572  001402                        BEQ     7$              ;BRANCH IF 0
3249  014574  104001                        ERROR                   ;ERROR! EXTRA DATA ENTRY
3250  014576  000470                        BR      16$             ;GO TO EXIT
3251
3252                                  ;TEST THAT THE DATA IS CORRECT IN ALL 16. ENTRIES
3253  014600  012701  001306          7$:   MOV     #TUMTAB,%1      ;GET FIST TT ADDRESS
3254  014604  012702  000020                MOV     #16.,%2         ;GET SCAN COUNT
3255  014610  005302                  8$:   DEC     %2              ;DECREMENT SCAN COUNT
3256  014612  100421                        BMI     10$             ;BRANCH IF 16. ENTRIES SCANNED
3257  014614  013737  016350  001302        MOV     OUTBUF,XMTDAT   ;GET TRANSMITTED DATA
3258  014622  043737  001304  001302        BIC     CARMSK,XMTDAT   ;CLEAR NON-TRANSMITTED BITS
3259  014630  113737  001306  001300        MOVB    TUMTAB,RCVDAT   ;GET RECEIVED DATA
3260  014636  123737  001302  001300        CMPB    XMTDAT,RCVDAT   ;COMPARE DATA
3261  014644  001402                        BEQ     9$
3262  014646  104011                        ERROR1                  ;ERROR  INCORRECT DATA
3263  014650  000443                        BR      16$             ;GO TO EXIT
3264  014652  005721                  9$:   TST     (1)+            ;INCREMENT TT ADDRESS
3265  014654  000755                        BR      8$              ;TEST NEXT ENTRY
3266
```

```
3267                                    ;CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3268  014656  012701  001306            10$:    MOV     #TUMTAB,%1      ;GET FIRST TT ADDRESS
3269  014662  012702  000020                    MOV     #16.,%2         ;GET SCAN COUNT
3270  014666  005302                    11$:    DEC     %2              ;DECREMENT SCAN COUNT
3271  014670  100403                            BMI     12$             ;BRANCH IF ALL LINES TESTED
3272  014672  042721  160777                    BIC     #160777,(1)+    ;CLEAR ALL BUT LINE NUMBER IN TT
3273  014676  000773                            BR      11$             ;DO NEXT TT ADDRESS
3274
3275                                    ;TEST THAT THERE IS AN ENTRY FOR EACH OF THE 16. LINES
3276  014700  005037  001302            12$:    CLR     XMTDAT
3277  014704  012701  000020                    MOV     #16.,%1
3278  014710  012702  000020            13$:    MOV     #16.,%2
3279  014714  012700  001306                    MOV     #TUMTAB,%0
3280  014720  023720  001302            14$:    CMP     XMTDAT,(0)+     ;TEST FOR LINE ENTRY
3281  014724  001406                            BEQ     15$             ;BRANCH IF FOUND
3282  014726  005302                            DEC     %2              ;DECREMENT SCAN COUNT
3283  014730  001373                            BNE     14$             ;LOOK AT NEXT ENTRY
3284  014732  005037  001300                    CLR     RCVDAT
3285  014736  104011                            ERROR1                  ;ERROR! NO ENTRY FOUND FOR THIS LINE
3286  014740  000407                            BR      16$             ;GO TO EXIT
3287  014742  005301                    15$:    DEC     %1              ;DECREMENT LINES FOUND COUNT
3288  014744  005701                            TST     %1
3289  014746  001404                            BEQ     16$             ;BRANCH IF ALL LINE TESTED
3290  014750  062737  001000  001302            ADD     #1000,XMTDAT    ;INCREMENT LINE NUMBER
3291  014756  000754                            BR      13$             ;GO DO NEXT LINE
3292  014760  104006                    16$:    SCOPE                   ;SCOPE
3293                                    ;*****************************************************************************
3294  014762  000136                    RT136:  136                     ;ROUTINE # 136                          *
3295  014764  015276                            RT137                   ;ADDR OF NEXT ROUTINE.                  *
3296  014766  000144                            100.                    ;ITERATION COUNT                        *
3297  014770  014772                            RT136A                  ;SCOPE ENTRY POINT.                     *
3298            000136                        X=X+1
3299                                    ;*****************************************************************************
3300
3301                                    ;TEST THAT THE DM11 CAN TRANSMIT A BREAK ON ALL LINES SIMULTANEOUSLY
3302  014772  013737  003642  015046  RT136A: MOV     @#TIME1,1$      ;GET TIME TO TRANSMIT ONE CHARACTER
3303  015000  005037  001306                    CLR     TUMTAB          ;CLEAR
3304  015004  004537  005510                    JSR     5,BMOVE         ;THE
3305  015010  001306                            TUMTAB                  ;TUMBLE
3306  015012  001307                            TUMTAB+1                ;TABLE
3307  015014  000177                            177
3308  015016  012777  000001  164224            MOV     #BIT0,@CSR      ;SET GO
3309  015024  012777  177777  164222            MOV     #-1,@BKCSR      ;SET BREAK BIT FOR ALL LINES
3310  015032  105777  164212                    TSTB    @CSR            ;WAIT FOR THE RECEIVER
3311  015036  100375                            BPL     .-4             ;TO RECEIVE A BREAK
3312  015040  005077  164210                    CLR     @BKCSR          ;CLEAR ALL BREAK BITS
3313  015044  104400                            DELAY                   ;WAIT ONE CHARACTER
3314  015046  000000                    1$:     OPEN                    ;TIME
3315  015050  022777  000201  164172            CMP     #201,@CSR       ;TEST THAT ONLY GO AND DONE ARE SET
3316  015056  001410                            BEQ     2$
3317  015060  017737  164164  001300            MOV     @CSR,RCVDAT     ;GET CSR ENTRY
3318  015066  012737  000201  001302            MOV     #201,XMTDAT     ;GET CORRECT RESULT
3319  015074  104011                            ERROR1                  ;ERROR! INCORRECT CSR DATA
3320  015076  000476                            BR      13$             ;EXIT
3321
3322                                    ;TEST THAT THERE IS 16. VALID DATA ENTRIES
```

```
3323  015100  012701  001306          2$:     MOV     #TUMTAB,%1      ;GET TUMBLE TABLE BASE ADDRESS
3324  015104  012702  000020                  MOV     #16.,%2         ;GET SCAN COUNT
3325  015110  005721                  3$:     TST     (1)+            ;TEST FOR VALID DATA ENTRY
3326  015112  100402                          BMI     4$              ;BRANCH IF VALID DATA ENTRY FOUND
3327  015114  104001                          ERROR                   ;ERROR! MISSING VALID DATA ENTRY
3328  015116  000466                          BR      13$             ;EXIT
3329  015120  005302                  4$:     DEC     %2              ;DECREMENT SCAN COUNT
3330  015122  001372                          BNE     3$              ;BRANCH IF 16. ENTRIES NOT SCANNED
3331
3332                                  ;TEST THAT THE BREAK BIT IS SET IN 16. TUMBLE TABLE ENTRIES
3333  015124  012701  001306                  MOV     #TUMTAB,%1
3334  015130  012702  000020                  MOV     #16.,%2
3335  015134  032721  040000          5$:     BIT     #BIT14,(1)+     ;BREAK BIT SET?
3336  015140  001002                          BNE     6$              ;BRANCH IF SET
3337  015142  104001                          ERROR                   ;ERROR! MISSING BREAK BIT
3338  015144  000453                          BR      13$             ;EXIT
3339  015146  005302                  6$:     DEC     %2              ;DECREMENT SCAN COUNT
3340  015150  001371                          BNE     5$
3341
3342                                  ;TEST THAT THE TUMBLE TABLE DATA BYTE IS ALL 0'S
3343  015152  012701  001306                  MOV     #TUMTAB,%1
3344  015156  012702  000020                  MOV     #16.,%2
3345  015162  105721                  7$:     TSTB    (1)+            ;TEST DATA BYTE
3346  015164  001402                          BEQ     8$              ;BRANCH IF 0'S
3347  015166  104001                          ERROR                   ;ERROR! INCORRECT DATA
3348  015170  000441                          BR      13$             ;EXIT
3349  015172  105721                  8$:     TSTB    (1)+            ;STEP TABLE POINTER TO NEXT DATA BYTE
3350  015174  005302                          DEC     %2
3351  015176  001371                          BNE     7$
3352
3353                                  ;CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3354  015200  012701  001306                  MOV     #TUMTAB,%1
3355  015204  012702  000020                  MOV     #16.,%2
3356  015210  042721  160777          9$:     BIC     #160777,(1)+    ;CLEAR ALL BUT LINE NUMBER
3357  015214  005302                          DEC     %2
3358  015216  001374                          BNE     9$
3359
3360                                  ;TEST THAT THERE IS A TUMBLE TABLE ENTRY FOR EACH LINE
3361  015220  005004                          CLR     %4              ;CLEAR LINE NUMBER
3362  015222  012703  000020                  MOV     #16.,%3
3363  015226  012702  000020          10$:    MOV     #16.,%2
3364  015232  012701  001306                  MOV     #TUMTAB,%1
3365  015236  020421                  11$:    CMP     %4,(1)+         ;TEST FOR LINR ENTRY FOR THIS LINE
3366  015240  001410                          BEQ     12$             ;BRANCH IF FOUND
3367  015242  005302                          DEC     %2
3368  015244  001374                          BNE     11$
3369  015246  010437  001302                  MOV     %4,XMTDAT
3370  015252  010437  001300                  MOV     %4,RCVDAT
3371  015256  104011                          ERROR1                  ;ERROR! NO LINE ENTRY FOUND FOR THIS LINE
3372  015260  000405                          BR      13$             ;EXIT
3373  015262  005303                  12$:    DEC     %3              ;ALL LINES BEEN FOUND
3374  015264  001403                          BEQ     13$             ;EXIT IF YES
3375  015266  062704  001000                  ADD     #1000,%4        ;SEARCH FOR
3376  015272  000755                          BR      10$             ;NEXT LINE
3377  015274  104006                  13$:    SCOPE                   ;SCOPE
3378                                  ;*****************************************************************************
```

```
3379  015276  000137                  RT137:  137                     ;ROUTINE # 137                          *
3380  015300  015314                          RT140                   ;ADDR OF NEXT ROUTINE.                  *
3381  015302  000002                          2                       ;ITERATION COUNT                        *
3382  015304  015306                          RT137A                  ;SCOPE ENTRY POINT.                     *
3383          000137                          X=X+1
3384                                  ;*******************************************************************************
3385
3386                                  ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
3387                                  ;NEXT LINE.
3388  015306  004537  005066          RT137A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
3389  015312  000040                          32.                     ;THIS MUCH BETWEEN LINES
3390                                  ;*******************************************************************************
3391  015314  000140                  RT140:  140                     ;ROUTINE # 140                          *
3392  015316  015332                          RT141                   ;ADDR OF NEXT ROUTINE.                  *
3393  015320  000002                          2                       ;ITERATION COUNT                        *
3394  015322  015324                          RT140A                  ;SCOPE ENTRY POINT.                     *
3395          000140                          X=X+1
3396                                  ;*******************************************************************************
3397
3398                                  ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
3399                                  ;NEXT LINE.
3400  015324  004537  005066          RT140A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
3401  015330  000020                          16.                     ;THIS MUCH BETWEEN LINES
3402                                  ;*******************************************************************************
3403  015332  000141                  RT141:  141                     ;ROUTINE # 141                          *
3404  015334  015350                          RT142                   ;ADDR OF NEXT ROUTINE.                  *
3405  015336  000002                          2                       ;ITERATION COUNT                        *
3406  015340  015342                          RT141A                  ;SCOPE ENTRY POINT.                     *
3407          000141                          X=X+1
3408                                  ;*******************************************************************************
3409
3410                                  ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
3411                                  ;NEXT LINE.
3412  015342  004537  005066          RT141A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
3413  015346  000010                          8.                      ;THIS MUCH BETWEEN LINES
3414                                  ;*******************************************************************************
3415  015350  000142                  RT142:  142                     ;ROUTINE # 142                          *
3416  015352  015366                          RT143                   ;ADDR OF NEXT ROUTINE.                  *
3417  015354  000002                          2                       ;ITERATION COUNT                        *
3418  015356  015360                          RT142A                  ;SCOPE ENTRY POINT.                     *
3419          000142                          X=X+1
3420                                  ;*******************************************************************************
3421
3422                                  ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
3423                                  ;NEXT LINE.
3424  015360  004537  005066          RT142A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
3425  015364  000004                          4                       ;THIS MUCH BETWEEN LINES
3426                                  ;*******************************************************************************
3427  015366  000143                  RT143:  143                     ;ROUTINE # 143                          *
3428  015370  015404                          RT144                   ;ADDR OF NEXT ROUTINE.                  *
3429  015372  000002                          2                       ;ITERATION COUNT                        *
3430  015374  015376                          RT143A                  ;SCOPE ENTRY POINT.                     *
3431          000143                          X=X+1
3432                                  ;*******************************************************************************
3433
3434                                  ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
```

```
                                        ;NEXT LINE.
015376  004537  005066                  RT143A: JSR     5,@#DLYXMT      ;GO DO TEST, DELAY
015402  000002                                  2                       ;THIS MUCH BETWEEN LINES
                                        ;*************************************************************************
015404  000144                          RT144:  144                     ;ROUTINE # 144                          *
015406  015422                                  RT145                   ;ADDR OF NEXT ROUTINE.                  *
015410  000002                                  2                       ;ITERATION COUNT                        *
015412  015414                                  RT144A                  ;SCOPE ENTRY POINT.                     *
        000144                                  X=X+1
                                        ;*************************************************************************

                                        ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
                                        ;NEXT LINE.
015414  004537  005066                  RT144A: JSR     5,@#DLYXMT      ;GO DO TEST, DELAY
015420  000001                                  1                       ;THIS MUCH BETWEEN LINES
                                        ;*************************************************************************
015422  000145                          RT145:  145                     ;ROUTINE # 145                          *
015424  015570                                  RT146                   ;ADDR OF NEXT ROUTINE.                  *
015426  000144                                  100.                    ;ITERATION COUNT                        *
015430  015432                                  RT145A                  ;SCOPE ENTRY POINT.                     *
        000145                                  X=X+1
                                        ;*************************************************************************


                                        ;TEST THAT THE DM11 WORKS PROPERLY WHEN THE HALF-DUPLEX BIT (CSR BIT 1)
                                        ;IS SET. THE TEST TRANSMITS DATA ON LINE 0, AND 'BREAKS' ON LINE 1. ONLY
                                        ;THE BREAK SHOULD BE RECEIVED ON LINE 0 IN THE TUMBLE TABLE.
015432  005037  001306                  RT145A: CLR     TUMTAB          ;CLEAR THE FIRST TWO
015436  005037  001310                          CLR     TUMTAB+2        ;TUMBLE TABLE ENTRIES
015442  012737  016350  001106                  MOV     #OUTBUF,CAT     ;SET UP TO
015450  012737  177777  001146                  MOV     #-1,WCT         ;TRANSMIT 1 CHARACTER
015456  012777  000007  163564                  MOV     #7,@CSR         ;SET GO, HALF DUPLEX & MAINT BITS
015464  012777  000001  163560                  MOV     #LBIT0,@BAR     ;TRANSMIT 1 CHAR. ON LINE 0
015472  012777  000002  163554                  MOV     #LBIT1,@GKCSR   ;SET BREAK ON LINE 1
015500  105777  163544                          TSTB    @CSR            ;WAIT FOR THE CHARACTER
015504  100375                                  BPL     .-4             ;TO BE RECEIVED
015506  005077  163542                          CLR     @BKCSR          ;CLEAR THE BREAK BIT ON LINE 1

                                        ;TEST THAT ONLY THE BREAK WAS RECEIVED
015512  022737  141000  001306                  CMP     #141000,TUMTAB  ;TDST FOR BREAK ENTRY (LINE 1)
015520  001410                                  BEQ     1$
015522  013737  001306  001300                  MOV     TUMTAB,RCVDAT   ;GET ACTUAL ENTRY
015530  012737  141000  001302                  MOV     #141000,XMTDAT  ;GET CORRECT ENTRY
015536  104011                                  ERROR1                  ;ERROR! INCORRECT BREAK ENTRY
015540  000407                                  BR      2$              ;GO TO EXIT
015542  013737  001310  001300  1$:     MOV     TUMTAB+2,RCVDAT ;TEST THAT NEXT ENTRY IS CLEAR
015550  001403                                  BEQ     2$              ;EXIT IF CORRECT
015552  005037  001302                          CLR     XMTDAT
015556  104011                                  ERROR1                  ;ERROR! SECOND ENTRY WAS NOT CLEAR
015560  005777  163464          2$:     TST     @CSR            ;WAIT FOR THE TRANSMITTER
015564  100375                                  BPL     .-4             ;TO FINISH
015566  104006                                  SCOPE                   ;SCOPE
                                        ;*************************************************************************
015570  000146                          RT146:  146                     ;ROUTINE # 146                          *
015572  177777                                  RTLAST                  ;ADDR OF NEXT ROUTINE.                  *
015574  000144                                  100.                    ;ITERATION COUNT                        *
015576  015600                                  RT146A                  ;SCOPE ENTRY POINT.                     *
```

```
3491                000146                          X=X+1
3492                                                ;*************************************************************************
3493
3494                                                ;TEST THAT THE DM11 RESPONDS CORRECTLY TO A RESET
3495  015600  012737  016350  001106  RT146A: MOV     #OUTBUF,CAT     ;SET UP TO TRANSMIT 10
3496  015606  012737  177770  001146           MOV     #-10,WCT        ;CHARACTERS ON LINE 0
3497  015614  013737  003642  015664           MOV     TIME1,2$        ;GET TIME TO TRANSMIT 2 CHARACTERS
3498  015622  013737  003642  015730           MOV     TIME1,6$
3499  015630  005037  001302                   CLR     XMTDAT
3500  015634  012777  000007  163406           MOV     #7,@CSR         ;SET MAINT., HALF DUPLEX & GO BITS
3501  015642  012777  000001  163402           MOV     #LBIT0,@BAR     ;START TO TRANSMIT ON LINE 0
3502  015650  012777  000002  163376           MOV     #LBIT1,@BKCSR   ;BREAK ON LINE 1
3503  015656  012704  000002                   MOV     #2,%4
3504  015662  104400                1$:        DELAY                   ;WAIT 2 CHARACTER
3505  015664  000000                2$:        OPEN                    ;TIMES
3506  015666  005304                           DEC     %4
3507  015670  001374                           BNE     1$
3508  015672  104005                           SRESET                  ;RESET
3509  015674  017737  163350  001300           MOV     @CSR,RCVDAT     ;GET CSR CONTENTS
3510  015702  001401                           BEQ     3$              ;BRANCH IF 0
3511  015704  104011                           ERROR1                  ;ERROR! CSR DID NOT CLEAR
3512  015706  017737  163340  001300  3$:      MOV     @BAR,RCVDAT     ;GET BAR CONTENTS
3513  015714  001402                           BEQ     4$              ;BRANCH IF 0
3514  015716  104011                           ERROR1                  ;ERROR! BAR IS NOT CLEAR
3515  015720  000427                           BR      9$              ;EXIT
3516  015722  012704  000010          4$:      MOV     #8.,%4
3517  015726  104400                5$:        DELAY                   ;WAIT 8 MORE CHARACTER TIMES
3518  015730  000000                6$:        OPEN
3519  015732  005304                           DEC     %4
3520  015734  001374                           BNE     5$
3521  015736  017737  163306  001300           MOV     @CSR,RCVDAT     ;TEST THAT CSR IS CLEAR
3522  015744  001402                           BEQ     7$
3523  015746  104011                           ERROR1                  ;ERROR! CSR WAS NOT CLEAR
3524  015750  000413                           BR      9$              ;GO TO EXIT
3525  015752  017737  163274  001300  7$:      MOV     @BAR,RCVDAT     ;TES THAT BAR IS CLEAR
3526  015760  001402                           BEQ     8$
3527  015762  104011                           ERROR1                  ;ERROR! BAR DID NOT CLEAR
3528  015764  000405                           BR      9$
3529  015766  017737  163262  001300  8$:      MOV     @BKCSR,RCVDAT   ;TEST THAT BKCSR IS CLEAR
3530  015774  001401                           BEQ     9$
3531  015776  104011                           ERROR1                  ;ERROR! BKCSR DID NOT CLEAR
3532  016000  104006                9$:        SCOPE                   ;SCOPE
```

```
3533                                            ;PRG1- TRANSMITTER SCOPE LOOP
3534   016002  000240                   PRG1:    NOP                     ;BEGIN
3535   016004  104000                            TYPE                    ;TYPE PROGRAM TITLE
3536   016006  017444                            PRG1M
3537   016010  004737  016166                    JSR     7,PARAM         ;GO GET USER PARAMETERS
3538   016014  004737  016240           PRG1R:   JSR     7,LOOP          ;GO LOOP TRANSMITTER
3539   016020  005777  163226           PRG1B:   TST     @BAR            ;WAIT FOR ALL LINES TO FINISH
3540   016024  001375                            BNE     PRG1B           ;BRANCH IF NOT DONE
3541   016026  005077  163216           PRG1C:   CLR     @CSR            ;CLEAR THE CSR
3542   016032  005037  016046                    CLR     PRG1D           ;CLEAR DELAY TIME
3543   016036  117737  163041  016046           MOVB    @SWR+1,PRG1D    ;GET DELAY
3544   016044  104400                            DELAY                   ;DELAY AS SPECIFIED
3545   016046  000000           PRG1D:   OPEN                            ;BY USER
3546   016050  000761                            BR      PRG1R           ;LOOP BACK
3547
3548
```

```
3549
3550                                        ;PRG2- RECEIVER SCOPE LOOP
3551  016052  000240                PRG2:   NOP                     ;BEGIN
3552  016054  104000                        TYPE                    ;TYPE PROGRAM
3553  016056  017474                        PRG2M                   ;TITLE
3554  016060  004737  016166                JSR     7,PARAM         ;GO GET USER PARAMETERS
3555  016064  004737  016240        PRG2R:  JSR     7,LOOP          ;GO START TRANSMITTER
3556  016070  012777  000001  163152 PRG2A:  MOV     #BIT0,@CSR      ;SET GO,CLEAR THE OTHERS
3557  016076  005777  163150        PRG2AA: TST     @BAR            ;HAVE ALL LINES SELECTED FINISHED
3558  016102  001415                        BEQ     PRG2B           ;BRANCH IF FINISHED TRANSMITTING
3559  016104  105777  163140                TSTB    @CSR            ;WAIT FOR THE RECEIVER TO
3560  016110  100372                        BPL     PRG2AA          ;RECEIVE A CHARACTER
3561  016112  042777  000200  163130        BIC     #BIT7,@CSR      ;CLEAR RECEIVER FLAG
3562  016120  020127  001504                CMP     %1,#TUMTAB+176  ;IS THE POINTER AT THE END OF THE TT
3563  016124  001002                        BNE     .+6             ;BRANCH IF NOT
3564  016126  012701  001304                MOV     #TUMTAB-2,%1    ;RESET POINTER
3565  016132  005721                        TST     (1)+            ;INCREMENT POINTER
3566  016134  000755                        BR      PRG2A           ;GO BACK & TEST TRANSMITTER FLAG
3567  016136  005077  163106        PRG2B:  CLR     @CSR            ;CLEAR THE CSR
3568  016142  005077  163104                CLR     @BAR            ;CLEAR THE BAR
3569  016146  005037  016162                CLR     PRG2C           ;CLEAR USER DELAY
3570  016152  117737  162725  016162        MOVB    @SWR+1,PRG2C    ;LOAD USER DELAY
3571  016160  104400                        DELAY                   ;DELAY AS SPECIFIED
3572  016162  000000                PRG2C:  OPEN                    ;BY USER
3573  016164  000737                        BR      PRG2R           ;REPEAT LOOP
3574
3575
3576                                        ;SUBROUTINE TO GET USER PARAMETERS (FOR PRG1 & 2)
3577  016166  104000                PARAM:  TYPE                    ;ASK USER WHICH LINE
3578  016170  017531                        LINPAR                  ;TO TEST
3579  016172  004537  003702                JSR     5,RECD          ;GET LINE AND PUT IT
3580  016176  000000                LINE:   0                       ;HERE
3581  016200  104000                PARAMA: TYPE                    ;ASK USER HOW MANY
3582  016202  017563                        HOWMAN                  ;CHARACTERS TO TRANSMIT
3583  016204  004537  003702                JSR     5,RECD          ;GET CHARS AND PUT IT
3584  016210  000000                CHARS:  0                       ;HERE
3585  016212  023727  016210  000310        CMP     CHARS,#200.     ;LIMIT RESPONSE TO 200.
3586  016220  101403                        BLOS    PARAMB          ;(CORE LIMITATION)
3587  016222  104000                        TYPE
3588  016224  017241                        M1
3589  016226  000764                        BR      PARAMA          ;RE-REQUEST PARAMETER
3590  016230  104000                PARAMB: TYPE                    ;TYPE INSTRUCTIONS
3591  016232  017356                        MM
3592  016234  104015                        CNTLU                   ;GO GET VALUE
3593  016236  000207                        RTS     7               ;EXIT
3594
3595
3596                                        ;SUBROUTINE TO TRANSMIT DATA FROM THE SR
3597  016240  117737  162636  016350 LOOP:   MOVB    @SWR,OUTBUF     ;FILL OUTPUT
3598  016246  004537  005510                JSR     5,BMOVE         ;BUFFER
3599  016252  016350                        OUTBUF                  ;WITH
3600  016254  016351                        OUTBUF+1                ;DATA TO BE
3601  016256  000307                        199.                    ;TRANSMITTED
3602  016260  012777  001106  162770        MOV     #CAT,@BAS%REG   ;INITIALIZE BASE REGISTER
3603  016266  012737  016350  001106        MOV     #OUTBUF,CAT     ;LOAD CURRENT
3604  016274  004537  005510                JSR     5,BMOVE         ;ADDRESS TABLE
```

```
3605  016300  001106                        CAT                             ;WITH ADDRESS
3606  016302  001110                        CAT+2                           ;OF OUTPUT BUFFER
3607  016304  000040                        32.
3608  016306  013737  016210  001146        MOV     CHARS,WCT               ;LOAD WORD COUNT
3609  016314  005437  001146                NEG     WCT                     ;FORM TWO'S COMPLEMENT
3610  016320  004537  005510                JSR     5,BMOVE                 ;TABLE WITH
3611  016324  001146                        WCT                             ;NUMBER OF
3612  016326  001150                        WCT+2                           ;CHARACTERS TO BE
3613  016330  000040                        32.                             ;TRANSMITTED
3614  016332  013737  016176  001272        MOV     LINE,LINBIT             ;SAVE LINES TO BE TRANSMITTED ON
3615  016340  013777  016176  162704        MOV     LINE,@BAR               ;START TRANSMITTING ON SELECTED LINES
3616  016346  000207                        RTS     7                       ;EXIT
3617
3618
3619  016350  000000                OUTBUF: 0                               ;FIRST ADDRESS OF 100.
3620          016514                        .=OUTBUF+100.                   ;CHARACTER OUTPUT BUFFER
3621  016514  000000                INBUF:  0                               ;FIRST ADDRESS OF 100.
3622          016660                        .=INBUF+100.                    ;CHARACTER INPUT BUFFER (WHERE RECEIVED
3623                                                                        ;DATA IS STORED)
3624
3625  016660  013746  000006        SUSWRR: MOV     @#6,-(SP)                       ;SAVE VECTORS
3626  016664  013746  000004                MOV     @#4,-(SP)
3627  016670  012737  016710  000004        MOV     #1$,@#4                 ;SET UP FOR TIMEOUT
3628  016676  022777  177777  162176        CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
3629  016704  001402                        BEQ     2$
3630  016706  000407                        BR      3$
3631  016710  022626                1$:     CMP     (SP)+,(SP)+             ;ADJUST STACK
3632  016712  012737  000176  001102  2$:     MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
3633  016720  012737  000174  001104        MOV     #DISPREG,DISPLAY                ;POINT TO SOFT DISPLAY REG
3634  016726  012637  000004        3$:     MOV     (SP)+,@#4               ;RESTORE VECTORS
3635  016732  012637  000006                MOV     (SP)+,@#6
3636  016736  000002                        RTI
3637
3638
3639                                ;ROUTINE TO CHECK FOR ↑G BEING TYPED
3640
3641  016740  022737  000176  001102  KBDINTT:CMP     #SWREG,SWR
3642  016746  001015                        BNE     1$
3643  016750  005037  017042                CLR     TMP1                    ;CLEAR TEMP AREA
3644  016754  117737  162616  017042        MOVB    @TKDBR,TMP1             ;FETCH THE BUFFER
3645  016762  142737  000200  017042        BICB    #200,TMP1               ;STRIP OFF PARITY
3646  016770  122737  000007  017042        CMPB    #7,TMP1                 ;WAS IT ↑G
3647  016776  001001                        BNE     1$                      ;NOP
3648  017000  104015                        CNTLU                           ;GO CHANGE IT
3649  017002  000002                1$:     RTI                             ;EXIT
3650
3651
3652                                ;ROUTINE TO CHANGE CONTENTS OF SWREG(LOC 176)
3653
3654  017004  022737  000176  001102  CNTLUU: CMP     #SWREG,SWR
3655  017012  001023                        BNE     FJX
3656  017014  104000                        TYPE
3657  017016  017106                        $SWREG
3658  017020  004537  005440                JSR     R5,OACNV                ;CONVERT TO ASCII
3659  017024  000176                        SWREG
3660  017026  017115                        $VALUE
```

```
3661   017030   000006                          6
3662   017032   104000                          TYPE
3663   017034   017115                          $VALUE
3664   017036   004537   003702                 JSR     5,RECO          ;GET TMP1 AND PUT IT
3665   017042   000000                 TMP1:    0                       ;HERE
3666   017044   022737   000007   004060        CMP     #7,CNT
3667   017052   001403                          BEQ     FJX
3668   017054   013777   017042   162020        MOV     TMP1,@SWR               ;CHANGE CONTENTS OF SWREG
3669   017062   000002                 FJX:     RTI
3670
```

```
                                        ;MESSAGES
017064 053045 041505 047524 WHERE:  .ASCII  '%VECTOR ADDRESS? @'
017072 020122 042101 051104
017100 051505 037523 040040
017106 051445 051127 020075 $SWREG: .ASCII  '%SWR= @'
017114    100
017115    040 020040 020040 $VALUE: .ASCII  '          NEW= @'
017122 020040 020040 047040
017130 053505 020075    100
017135    045 020075        $CTLU:  .ASCII  '%= '
017140 052445 044516 020124 WHICH:  .ASCII  '%UNIT #(8)? @'
017146 024043 024470 020077
017154    100
017155    045 044103 051101 LEVEL:  .ASCII  '%CHAR LENGTH @'
017162 046040 047105 052107
017170 020110    100
017173    045 051105 020122 ERDAT:  .ASCII  '%ERR S/B: '
017200 027523 035102    040
017205    040 020040 020040 AASB:   .ASCII  '       WAS: '
017212 020040 040527 035123
017220    040
017221    040 020040 020040 AWAS:   .ASCII  '      @'
017226 040040
017230 050045 043522 020056 M0:     .ASCII  '%PRG. # @'
017236 020043    100
017241    045 040077        M1:     .ASCII  '%?@'
017244 052045 051505 020124 M2:     .ASCII  '%TEST DZDMA COMPLETE@'
017252 055104 046504 020101
017260 047503 050115 042514
017266 042524    100
017271    045 042523 020124 M3:     .ASCII  '%SET SR OPTIONS. NORMAL OPERATION'
017276 051123 047440 052120
017304 047511 051516 020056
017312 047516 046522 046101
017320 047440 042520 040522
017326 044524 047117
017332 051511 051440 036522       .ASCII  'IS SR=0.PRESS CONT.@'
017340 027060 051120 051505
017346 020123 047503 052116
017354 040056
017356 050045 052125 041440 M4:     .ASCII  '%PUT CHAR IN SR (0-7);DELAY IN SR (8-15)@'
017364 040510 020122 047111
017372 051440 020122 030050
017400 033455 035451 042504
017406 040514 020131 047111
017414 051440 020122 034050
017422 030455 024465    100
017427    045 047514 044507 PRGOM:  .ASCII  '%LOGIC TESTS@'
017434 020103 042524 052123
017442 040123
017444 052045 040522 051516 PRG1M:  .ASCII  '%TRANSMITTER SCOPE LOOP@'
017452 044515 052124 051105
017460 051440 047503 042520
017466 046040 047517 040120
017474 052045 040522 051516 PRG2M:  .ASCII  '%TRANSMIT/RECEIVE SCOPE LOOP@'
017502 044515 027524 042522
```

```
3727  017510  042503  053111  020105
3728  017516  041523  050117  020105
3729  017524  047514  050117     100
3730  017531     045  054524  042520  LINPAR: .ASCII  '%TYPE LINES TO BE TESTED @'
3731  017536  046040  047111  051505
3732  017544  052040  020117  042502
3733  017552  052040  051505  042524
3734  017560  020104     100
3735  017563     045  047510  020127  HOWMAN: .ASCII  '%HOW MANY CHARACTERS @'
3736  017570  040515  054516  041440
3737  017576  040510  040522  052103
3738  017604  051105  020123     100
3739  017611     045  020122          EMO:    .ASCII  '%R '
3740  017614  020040  050040  036503  ATNUMB: .ASCII  '   PC= '
3741  017622     040
3742  017623     040  020040  020040  APC:    .ASCII  '      @'
3743  017630  040040
3744  017632     015     012          MSG1:   .BYTE   15,12
3745  017634  044124  020105  052521          .ASCII  'THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
3746  017642  041511  020113  051102
3747  017650  053517  020116  047506
3748  017656  020130  052512  050115
3749  017664  042105  047440  042526
3750  017672  020122  044124  020105
3751  017700  040514  054532  042040
3752  017706  043517  020123  040502
3753  017714  045503  030440  031462
3754  017722  032464  033466  034470
3755  017730     060
3756          000001                  .END
```

```
A     = 000000          535#  2321#  2835#  3013#
AASB    017205          815   3689#
APC     017623          786   3742#
APCADD  001614          751#   776*   779*   785
AREA    013014         2669   2680   2699*  2700   2710   2720#
ATNUMB  017614          790   3740#
AWAS    017221          819   3692#
A12K    013034         2729#
A16K    013040         2731#
A20K    013044         2733#
A24K    013050         2735#
A28K    013054         2737#
A8K     013024         2724#
BAR     001252          699#   692*  1108*  1225   1415   1503*  1988   2016*  2018   2022*  2023*  2024   2061*
                       2062*  2063   2066   2069*  2523*  2527   2531*  2569*  2572*  2633*  2672*  2703*  2767
                       2769   2781   2783   2813*  3224   3228*  3466*  3501*  3512   3525   3539   3557   3568*
                       3615*
BARDAT  001274          708#
BASREG  001256          701#  1014*  1500*  1931*  1932   1940*  1941   1962   1966*  1967   1973*  1974*  1989*
                       2565*  2628*  3602*
BAT     001206          695#   696
BELL  = 000007          530#   801
BFRPTR  001606          748#
BIT0  = 000001          478#  1109   1285   1397   1524   1678   1679   1683   2812   3214   3308   3556
BIT1  = 000002          477#  1697   1698   1702
BIT10 = 002000          468#
BIT11 = 004000          467#   900
BIT12 = 010000          466#  1241   1249   1792   1793   1797   2110   2538
BIT13 = 020000          465#   782   1312   1342   1527   1532   1548   1811   1812   1816   2111   3231
BIT14 = 040000          464#   897   1373   1527   1532   1861   1869   1895   1936   1944   1971   2533   2573
                       2645   3335
BIT15 = 100000          446    463#  1249   1288   1316   1346   1527   1536   2537
BIT2  = 000004          476#  1109   1716   1717   1721   2808
BIT3  = 000010          475#
BIT4  = 000020          474#  1735   1736   1740
BIT5  = 000040          473#  1754   1755   1759
BIT6  = 000100          472#  1109   1298   1306   1773   1774   1778
BIT7  = 000200          471#  1306   1335   1364   1367   1539   2757   2816   3561
BIT8  = 000400          470#   797
BIT9  = 001000          469#   875    904
BKCSR   001254          700#   894*  1361*  1368*  1856*  1857   1865*  1866   1887   1890*  1891   1897*  1898*
                       1913*  1916   2085*  2086*  2087   2090   2093*  3309*  3312*  3467*  3470*  3502*  3529
BMOVE   005510         1391   1444   1485#  1519   3207   3304   3598   3604   3610
BRKTST  004742         1360#  2845   2856   2867   2878   2889   2900   2911   2922   2933   2944   2955   2966
                       2977   2988   2999   3010
BRK0    013472         2841   2845#
BRK1    013510         2852   2856#
BRK10   013652         2929   2933#
BRK11   013670         2940   2944#
BRK12   013706         2951   2955#
BRK13   013724         2962   2966#
BRK14   013742         2973   2977#
BRK15   013760         2984   2988#
BRK16   013776         2995   2999#
BRK17   014014         3006   3010#
BRK2    013526         2863   2867#
```

DZDMAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
BRK3     013544          2874    2878#
BRK4     013562          2885    2889#
BRK5     013600          2896    2900#
BRK6     013616          2907    2911#
BRK7     013634          2918    2922#
CARMSK   001304           712#    1091*   1314    1344    1583    2681    2711    3258
CAT      001106           691#     692    1014    1105*   1236    1500    1501*   1989    2514    2628    2629*   2639    2643
                         2669*   2700*   3463*   3495*   3602    3603*   3605    3606
CC     = 177776           441#
CHALT  = 104003           518#
CHARS    016210          3584#   3585    3608
CLKINT   001260           702#    1042*   1106*   1297*
CLKLVL   001262           703#    1107*
CNT      004060          1155*   1172*   1191#   3666
CNTLU  = 104015           528#     869    3592    3648
CNTLUU   017004           742     3654#
CNVDAT   002034           768      780     813#
COUNT    001604           747#    1111*   1118*   1389*   1401    1511*   1515*
CSR      001250           698#     893*   1060    1067    1109*   1114*   1121*   1228    1241*   1249*   1250    1285*   1286
                         1288*   1290    1298*   1306*   1307    1331    1333    1335*   1360*   1362    1364*   1365    1367*
                         1397*   1419    1422    1524*   1527    1529    1532    1536*   1537    1539*   1662    1678*   1679
                         1683*   1684    1697*   1698    1702*   1703    1716*   1717    1721*   1722    1735*   1736    1740*
                         1741    1754*   1755    1759*   1760    1773*   1774    1778*   1779    1792*   1793    1797*   1798
                         1811*   1812    1816*   1817    1831*   1834    1838*   1839*   1840    2037*   2038*   2039    2042
                         2045*   2049    2073    2097    2110*   2111*   2129*   2137*   2138*   2522*   2533    2537*   2538*
                         2547*   2573    2597    2599*   2600    2632*   2634    2636*   2637    2670*   2673    2675    2677*
                         2701*   2704    2706    2708*   2750*   2755    2757*   2761    2777    2808*   2812*   2814    2816*
                         2827    2829    3214*   3229*   3231    3234*   3238*   3308*   3310    3315    3317    3465*   3468
                         3483    3500*   3509    3521    3541*   3556*   3559    3561*   3567*
CURTST   001510           717#     877     883     922*
DATCHK=  104002           517#    1585
DATTST   005576          1510#   3023    3034    3045    3056    3067    3078    3089    3100    3111    3122    3133    3144
                         3155    3166    3177    3188
DAT0     014032          3019    3023#
DAT1     014050          3030    3034#
DAT1AA   005602          1511#
DAT10    014212          3107    3111#
DAT11    014230          3118    3122#
DAT12    014246          3129    3133#
DAT13    014264          3140    3144#
DAT14    014302          3151    3155#
DAT15    014320          3162    3166#
DAT16    014336          3173    3177#
DAT17    014354          3184    3188#
DAT2     014066          3041    3045#
DAT3     014104          3052    3056#
DAT4     014122          3063    3067#
DAT5     014140          3074    3078#
DAT6     014156          3085    3089#
DAT7     014174          3096    3100#
DELAY  = 104400           529#    1402    1411    2525    2553    2570    2578    2610    2775    3222    3313    3504    3517
                         3544    3571
DISPLA   001104           690#    3633*
DISPRE   000174           676#    3633
DLY      005264           551     1428#
DLYXMT   005066          1389#   3388    3400    3412    3424    3436    3448
```

DZDMAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
DMPAR   003126            844    1024#   1037
DMPARB  003242           1047#   1055
DMPARC  003356           1074#   1085
DTCHK   001640            731     766#
EHALT = 104004            519#
EMO     017611            793    3739#
EMTINT  002542            549     927#
EMTTAB  001540            729#    933
ERDAT   017173            796    3687#
ERR     001660            730     775#
ERRA    001716            777     781#
ERRB    001760           775*    778*    794#
ERRC    001766            783     797#
ERRD    002020            798     804#
ERREX   002030            805     807#
ERROR = 104001            516#   1116    1200    1230    1245    1252    1294    1302    1309    1371    1375    1534    1542
                         1603    1607    1667    1681    1686    1700    1705    1719    1724    1738    1743    1757    1762
                         1776    1781    1795    1800    1814    1819    2114    2132    2141    2535    2575    2779    2833
                         3233    3245    3249    3327    3337    3347
ERROR1= 104011            524#    769    1227    1234    1239    1321    1327    1339    1350    1382    1417    1423    1560
                         1836    1842    1860    1868    1894    1918    1935    1943    1970    2002    2020    2026    2043
                         2050    2067    2074    2091    2098    2530    2544    2604    2644    2684    2714    2763    2770
                         2785    2821    3227    3262    3285    3319    3371    3477    3482    3511    3514    3523    3527
                         3531
ERRVEC= 000004            443#    833*    840*    871*   1661*   1663*   2566*   2580*   2626*   2627*   2653*   2654*   2694*
                         2717#
ERR1    001676            738     778#
ESCOPE  002364            735     890#
FJX     017062           3655    3667    3669#
FORWD   002510            874     917#
FORWDA  002534            922#
GETRDY  002262            868     870#    887     905     911    1651
GTBIN   005370           1454#   1513
GTLINB  006330           1215    1499    1633#
GTRDYA  002302            874#    885
GTRDYC  002324            876     879#
GTRDYD  002346            882     884#
GTRDYX  002270            871#    907
HOWMAN  017563           3582    3735#
ICTR    001516            720#    902*    920*
INBIN   005340           1443#
INBUF   016514           1523    1575    3621#   3622
INCRTN  001632            761#    886
INIT    003076            739    1014#
INITIA= 104012            525#    846     860     873     895
KBDIN = 104014            527#    781     807     891
KBDINT  016740            741    3641#
KSTART  001506            716#    870    1646*   1649*
LBIT0 = 000001            462#   1108    1395    2520    2569    2633    2672    2703    2813    2846    3213    3466    3501
LBIT1 = 000002            461#   2857    3467    3502
LBIT10= 000400            454#   2934
LBIT11= 001000            453#   2945
LBIT12= 002000            452#   2956
LBIT13= 004000            451#   2967
LBIT14= 010000            450#   2978
LBIT15= 020000            449#   2989
```

```
LBIT16= 040000          448#   3000
LBIT17= 100000          447#   3011
LBIT2 = 000004          460#   2868
LBIT3 = 000010          459#   2879
LBIT4 = 000020          458#   2890
LBIT5 = 000040          457#   2901
LBIT6 = 000100          456#   2912
LBIT7 = 000200          455#   2923
LENGTH  003366         1077#   1078    1080*   1081    1086    1088*   1089*   1090
LENOKA  003404         1079    1081#
LENOKB  003414         1083#   1087
LENOKC  003422         1082    1086#
LENOKD  003462         1091    1097#
LEVEL   017155         1075    3684#
LINBIT  001272          707#   1395*   1407*   1503    1634*   1637*   2602    2603    2607*   2767    2788*   3213*   3219*
                       3614#
LINE    016176         1214#   1231    1282*   1315    1345    1396*   1406*   1498    1510*   1556    1558    1568    1635
                       2594#   2606*   2749*   2752    2787*   3215*   3580#   3614    3615
LINE0 = 000000          490#   2153    2331    3024
LINE1 = 000002          491#   2164    2342    3035
LINE10= 000020          498#   2241    2419    3112
LINE11= 000022          499#   2252    2430    3123
LINE12= 000024          500#   2263    2441    3134
LINE13= 000026          501#   2274    2452    3145
LINE14= 000030          502#   2285    2463    3156
LINE15= 000032          503#   2296    2474    3167
LINE16= 000034          504#   2307    2485    3178
LINE17= 000036          505#   2318    2496    3189
LINE2 = 000004          492#   2175    2353    3046
LINE3 = 000006          493#   2186    2364    3057
LINE4 = 000010          494#   2197    2375    3068
LINE5 = 000012          495#   2208    2386    3079
LINE6 = 000014          496#   2219    2397    3090
LINE7 = 000016          497#   2230    2408    3101
LINPAR  017531         3578    3730#
LOOP    016240         3538    3555    3597#
LTST0   010740         2148    2152#
LTST1   010756         2159    2163#
LTST10  011120         2236    2240#
LTST11  011136         2247    2251#
LTST12  011154         2258    2262#
LTST13  011172         2269    2273#
LTST14  011210         2280    2284#
LTST15  011226         2291    2295#
LTST16  011244         2302    2306#
LTST17  011262         2313    2317#
LTST2   010774         2170    2174#
LTST3   011012         2181    2185#
LTST4   011030         2192    2196#
LTST5   011046         2203    2207#
LTST6   011064         2214    2218#
LTST7   011102         2225    2229#
MACHER  000004          539#
MANUAL= 100000          446#
MSG1    017632         1392    3744#
MO      017230          848    3694#
```

```
M1        017241              762   1036   1054   1084   1187   3588   3696#
M2        017244              909   3697#
M3        017271              866   3701#
M4        017356             3591   3711#
NOP     = 000240              444#
NXTST     001514              719#   870*   884    906    917    919*
OACNV     005440              784    788    813    817   1466#  3658
OACNVA    005452             1471#  1480
OPEN    = 000000              445#   691    693    695    697    702    704    706    707    708    709    710    711
                              712    714    716    717    718    719    720    721    722    747    748    749    750
                              751    752    753    794    947    948    962    963   1009   1129   1130   1439   1440
                             1449   1450   1451   1624   1625   1626   1627   1628   1629   1630   2526   2554   2571
                             2579   2776   3223   3314   3505   3518   3545   3572
OUTBUF    016350             1105   1235   1278*  1393   1501   1512   1574   1580   3211*  3257   3463   3495   3597*
                             3599   3600   3603   3619#  3620
OVRLAY    003646             1025   1133#
PARAM     016166             3537   3554   3577#
PARAMA    016200             3581#  3589
PARAMB    016230             3586   3590#
PARBIT    001610              749#
PAR0      006312             1613*  1624#
PAR1      006314             1614*  1625#
PAR2      006316             1615*  1626#
PAR3      006320             1616*  1627#
PAR4      006322             1617*  1628#
PAR5      006324             1618*  1629#
PAR6      006326             1619*  1630#
PC      =X000007              513#
PCADD     001612              750#   776    779    929*
PFAIL     004062              853    857   1194#
POPSP   = 005726              479#  1424   1609
POPSP2  = 022626              480#  1126   2716
PRGLIM    001522              722#   851
PRGNUM    002146              850#   851*   852*   855    861
PRGTAB    001524              723#   856
PRG0      006362              723   1644#
PRG0M     017427             1645   3718#
PRG0R     006404              726   1649#
PRG1      016002              724   3534#
PRG1B     016020             3539#  3540
PRG1C     016026             3541#
PRG1D     016046             3542*  3543*  3545#
PRG1M     017444             3536   3721#
PRG1R     016014              727   3538#  3546
PRG2      016052              725   3551#
PRG2A     016070             3556#  3566
PRG2AA    016076             3557#  3560
PRG2B     016136             3558   3567#
PRG2C     016162             3569*  3570*  3572#
PRG2M     017474             3553   3725#
PRG2R     016064              728   3555#  3573
PRTY0   = 000000              488#
PRTY1   = 000040              487#
PRTY2   = 000100              486#
PRTY3   = 000140              485#
PRTY4   = 000200              484#  2130
```

DZDMAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
PRTY5 = 000240          483#   703    705
PRTY6 = 000300          482#
PRTY7 = 000340          481#   550    552    854    858   1015   1107   1115   1244   1248   1281   1301   1304
                       2127   2541   2548   2627
PRVCNT  001616           752#  1887#  1898   1962#  1974
PSW   = 177776           442#  1015#  1110#  1115#  1242#  1244#  1248#  1281#  1299#  1301#  1304#  2112#  2127#
                        2130#  2135#  2539#  2541#  2548#
PTO     005364          1450#  1454   1460#
PT1     005366          1451#  1454#  1455#  1458#  1459#  1460   1461
PWRUP   004072          1194   1198#
RCVDAT  001300           710#   766    818   1225#  1232#  1236#  1237   1311#  1312#  1317   1324#  1336#  1341#
                        1342#  1347   1379#  1380#  1415#  1422#  1559#  1578#  1584#  1590#  1597#  1834#  1840#
                        1857#  1858   1866#  1891#  1992   1916#  1932#  1933   1941#  1967#  1968   2001#  2018#
                        2024#  2042#  2049#  2066#  2073#  2090#  2097#  2527#  2543#  2602#  2643#  2678#  2679#
                        2682   2709#  2712   2751#  2758#  2759   2783#  2810#  2817#  3224#  3259#  3260   3284#
                        3317#  3370#  3475#  3479#  3509#  3512#  3521#  3525#  3529#
RCVTST  004320          1278#  2330   2341   2352   2363   2374   2385   2396   2407   2418   2429   2440   2451
                        2462   2473   2484   2495
RCV0    011300          2326   2330#
RCV1    011316          2337   2341#
RCV10   011460          2414   2418#
RCV11   011476          2425   2429#
RCV12   011514          2436   2440#
RCV13   011532          2447   2451#
RCV14   011550          2458   2462#
RCV15   011566          2469   2473#
RCV16   011604          2480   2484#
RCV17   011622          2491   2495#
RCV2    011334          2348   2352#
RCV3    011352          2359   2363#
RCV4    011370          2370   2374#
RCV5    011406          2381   2385#
RCV6    011424          2392   2396#
RCV7    011442          2403   2407#
RECD    003702           849   1028   1049   1076   1153#  3579   3583   3664
RIND    005362          1443#  1445   1446   1449#  1456#
RNGEN   002712           973#  1888   1963
RORPAR  006250          1551   1577   1599   1613#  1621
RP1     002760           973    977#   984    986#
RP2     002762           976    980    983#   987#
RSTART  001532           726#   862
RSTAT1  002130           681    845#
RSTAT2  002206           683    857#  1201
RSTPC   002670           951#   960    962#
RSTPSW  002672           952#   959    963#
RSTREG= 104010           523#  1481   1492
RSTRG   002634           737    951#
RTLAST= 177777           531#  3480
RTNNO   001512           718#   789    881    918#  1647#  1650#
RTO     006422          1646   1649   1653#
RTOA    006432          1656   1661#
RT1     006464          1654   1670#
RT1A    006474          1673   1678#
RT10    007132          1785   1803#
RT10A   007142          1806   1811#
RT100   013534          2861   2871#
```

RT101	013552	2872	2882#	
RT102	013570	2883	2893#	
RT103	013606	2894	2904#	
RT104	013624	2905	2915#	
RT105	013642	2916	2926#	
RT106	013660	2927	2937#	
RT107	013676	2938	2948#	
RT11	007204	1804	1822#	
RT11A	007214	1825	1831#	1837
RT110	013714	2949	2959#	
RT111	013732	2960	2970#	
RT112	013750	2971	2981#	
RT113	013766	2982	2992#	
RT114	014004	2993	3003#	
RT115	014022	3004	3016#	
RT116	014040	3017	3027#	
RT117	014056	3028	3038#	
RT12	007274	1823	1846#	
RT12A	007304	1849	1855#	
RT120	014074	3039	3049#	
RT121	014112	3050	3060#	
RT122	014130	3061	3071#	
RT123	014146	3072	3082#	
RT124	014164	3083	3093#	
RT125	014202	3094	3104#	
RT126	014220	3105	3115#	
RT127	014236	3116	3126#	
RT13	007430	1847	1878#	
RT13A	007440	1881	1886#	
RT130	014254	3127	3137#	
RT131	014272	3138	3148#	
RT132	014310	3149	3159#	
RT133	014326	3160	3170#	
RT134	014344	3171	3181#	
RT135	014362	3182	3192#	
RT135A	014372	3195	3206#	
RT136	014762	3193	3294#	
RT136A	014772	3297	3302#	
RT137	015276	3295	3379#	
RT137A	015306	3382	3388#	
RT14	007542	1879	1904#	
RT14A	007552	1907	1913#	
RT140	015314	3380	3391#	
RT140A	015324	3394	3400#	
RT141	015332	3392	3403#	
RT141A	015342	3406	3412#	
RT142	015350	3404	3415#	
RT142A	015360	3418	3424#	
RT143	015366	3416	3427#	
RT143A	015376	3430	3436#	
RT144	015404	3428	3439#	
RT144A	015414	3442	3448#	
RT145	015422	3440	3451#	
RT145A	015432	3454	3461#	
RT146	015570	3452	3487#	
RT146A	015600	3490	3495#	

RT15	007602	1905	1921#
RT15A	007612	1924	1930#
RT16	007736	1922	1953#
RT16A	007746	1956	1961#
RT17	010054	1954	1980#
RT17A	010064	1983	1988#
RT2	006536	1671	1689#
RT2A	006546	1692	1697#
RT20	010146	1981	2007#
RT20A	010156	2010	2015#
RT21	010234	2008	2029#
RT21A	010244	2032	2037#
RT22	010342	2030	2053#
RT22A	010352	2056	2061#
RT23	010450	2054	2077#
RT23A	010460	2080	2085#
RT24	010556	2078	2101#
RT24A	010566	2104	2109#
RT25	010622	2102	2119#
RT25A	010632	2122	2127#
RT26	010730	2120	2145#
RT27	010746	2146	2156#
RT3	006610	1690	1708#
RT3A	006620	1711	1716#
RT30	010764	2157	2167#
RT31	011002	2168	2178#
RT32	011020	2179	2189#
RT33	011036	2190	2200#
RT34	011054	2201	2211#
RT35	011072	2212	2222#
RT36	011110	2223	2233#
RT37	011126	2234	2244#
RT4	006662	1709	1727#
RT4A	006672	1730	1735#
RT40	011144	2245	2255#
RT41	011162	2256	2266#
RT42	011200	2267	2277#
RT43	011216	2278	2288#
RT44	011234	2289	2299#
RT45	011252	2300	2310#
RT46	011270	2311	2323#
RT47	011306	2324	2334#
RT5	006734	1728	1746#
RT5A	006744	1749	1754#
RT50	011324	2335	2345#
RT51	011342	2346	2356#
RT52	011360	2357	2367#
RT53	011376	2368	2378#
RT54	011414	2379	2389#
RT55	011432	2390	2400#
RT56	011450	2401	2411#
RT57	011466	2412	2422#
RT6	007006	1747	1765#
RT6A	007016	1768	1773#
RT60	011504	2423	2433#
RT61	011522	2434	2444#

```
RT62     011540            2445    2455#
RT63     011556            2456    2466#
RT64     011574            2467    2477#
RT65     011612             835*   2478    2488#
RT66     011632            2489    2500#
RT66A    011646            2503    2512#
RT67     012072            2501    2557#
RT67A    012102            2560    2565#
RT7      007060            1766    1784#
RT7A     007070            1787    1792#
RT70     012202             835    2558    2583#
RT70A    012212            2586    2594#
RT71     012316            2584    2614#
RT71A    012326            2617    2623#
RT72     012512            2615    2657#
RT72A    012522            2660    2667#
RT73     013060            2658    2740#
RT73A    013070            2743    2749#
RT74     013274            2741    2792#
RT74A    013304            2795    2802#
RT75     013462            2793    2838#
RT76     013500            2839    2849#
RT77     013516            2850    2860#
R0     =%000000             506#   1153    1158*   1159*   1160    1161    1163    1165*   1166    1171    1184*
R1     =%000001             507#
R2     =%000002             508#
R3     =%000003             509#
R4     =%000004             510#
R5     =%000005             511#   1183    1185*   3658*
SAVREG= 104007              522#   1466    1485
SAVRG   002574              736     937#
SCOPE = 104006              521#   1254    1351    1383    1425    1610    1664    1687    1706    1725    1744    1763    1782
                           1801    1820    1844    1876    1902    1919    1951    1978    1999    2027    2051    2075    2099
                           2117    2143    2555    2581    2612    2655    2718    2790    2834    3292    3377    3485    3532
SCOPEA   002422             899#    903
SCOPEB   002426             898     900#
SCOPED   002444             878     901     904#
SCOPTR   001520             721#    896     921*
SP     =%000006             512#    896*    899    1016    1017*   1018    1153*   1184    1289*   1292*   1296    1497*   1504
                           1620*   1633*   1640    3625*   3626*   3631    3674    3635
SPBOT    001100             687#    831     838     845     859    1016*   1017    1199
SRESET= 104005              520#   1833    1915    2017    3508
SRSET    002240             863#   1648
SRSETT   002674             734     966#    968*
START    002062             680     831#
SUSWR = 104013              526#    832
SUSWRR   016660             740    3625#
SVRPC    002630             937*    945     947#
SVRPSW   002632             938*    944     948#
SWR      001102             689#    782     797     804     863     875     879     897     900     904    1861    1869    1895
                           1936    1944    1971    2645    3543    3570    3597    3628    3632*   3641    3654    3668*
SWREG    000176             677#    863    3632    3641    3654    3659
TIME     001620             753#
TIMEA    003546            1111#   1120
TIMEB    003602            1113    1118#
TIMEC    003612            1106    1121#
```

DZDMAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
TIMER   003472          1103#  1117   2512
TIME1   003642          1103*  1112*  1122   1129#  1400   1410   2552   2577   2609   3221   3302   3497   3498
TIME14  003644          1122*  1123*  1125*  1130#  2524   2568   2774
TKCSR   001574           743#  1156
TKDBR   001576           744#  1158   3644
TMPI    017042          3643*  3644*  3645*  3646   3665#  3668
TPCSR   001600           745#   799    802   1001   1175   1178   1181
TPDBR   001602           746#   801*  1000*  1160*  1177*  1180*
TTDAT   001270           706#  1554*  1555*  1556   1559
TTPTR   001276           709#
TUMTAB  001306           714#   715   1279*  1280*  1311   1323*  1324   1336   1341   1369   1373   1377   1380
                        1517   1518*  1520   1521   1563   1565   2679   2709   2802   2807   2831   3206*  3208
                        3209   3240   3253   3259   3268   3279   3303*  3305   3306   3323   3333   3343   3354
                        3364   3461*  3462*  3473   3475   3479   3562   3564
TYP     002764           729    989#
TYPA    002774           992#   999   1008
TYPC    003012           994    996#
TYPD    003030           998   1000#  1005   1007
TYPDAT  003074           992*   993    996   1000   1004*  1006*  1009#
TYPE  = 104000           515#   761    792    795    847    865    908   1026   1035   1047   1053   1074   1083
                        1186   1188   1644   3535   3552   3577   3581   3587   3590   3656   3662
TYPF    003046           997   1004#
TYPG    003060          1006#
UNIT    003252          1050#  1051   1056*  1057*  1058*  1062
UNTOKA  003272          1052   1056#
UNTOKB  003316          1061#  1064
UNTOKC  003346          1069#  1072
VAC     001246           697#   713
VECOK   003162          1031   1033#
VECOKA  003172          1035#  1039   1041
VECOKB  003200          1034   1038#
VECTOR  003144          1029#  1030   1032*  1033   1038   1040   1042   1043*  1044
WCT     001146           693#   694   1104*  1232   1502*  1569   2631*  2671*  2702*  2811*  3464*  3496*  3608*
                        3609*  3611   3612
WHERE   017064          1027   3672#
WHICH   017140          1048   3681#
X     = 000146           534#  1652   1657#  1669   1674#  1688   1693#  1707   1712#  1726   1731#  1745   1750#
                        1764   1769#  1783   1788#  1802   1807#  1821   1826#  1845   1850#  1877   1882#  1903
                        1908#  1920   1925#  1952   1957#  1979   1984#  2006   2011#  2028   2033#  2052   2057#
                        2076   2081#  2100   2105#  2118   2123#  2144   2149#  2155   2160#  2166   2171#  2177
                        2182#  2188   2193#  2199   2204#  2210   2215#  2221   2226#  2232   2237#  2243   2248#
                        2254   2259#  2265   2270#  2276   2281#  2287   2292#  2298   2303#  2309   2314#  2322
                        2327#  2333   2338#  2344   2349#  2355   2360#  2366   2371#  2377   2382#  2388   2393#
                        2399   2404#  2410   2415#  2421   2426#  2432   2437#  2443   2448#  2454   2459#  2465
                        2470#  2476   2481#  2487   2492#  2498   2499   2504#  2556   2561#  2582   2587#  2613
                        2618#  2656   2661#  2739   2744#  2791   2796#  2837   2842#  2848   2853#  2859   2864#
                        2870   2875#  2881   2886#  2892   2897#  2903   2908#  2914   2919#  2925   2930#  2936
                        2941#  2947   2952#  2958   2963#  2969   2974#  2980   2985#  2991   2996#  3002   3007#
                        3015   3020#  3026   3031#  3037   3042#  3048   3053#  3059   3064#  3070   3075#  3081
                        3086#  3092   3097#  3103   3108#  3114   3119#  3125   3130#  3136   3141#  3147   3152#
                        3158   3163#  3169   3174#  3180   3185#  3191   3196#  3293   3298#  3378   3383#  3390
                        3395#  3402   3407#  3414   3419#  3426   3431#  3438   3443#  3450   3455#  3486   3491#
XMITD   005532          1217   1283   1329   1398   1497#  1525   2595   2753   3216
XMTDAT  001302           711#   766    814   1216*  1235*  1237   1313*  1314*  1315*  1316*  1317   1326*  1338*
                        1343*  1344*  1345*  1346*  1347   1381*  1390*  1421*  1558*  1579*  1582*  1583*  1832*
                        1855*  1856   1858   1863   1864*  1871*  1872   1874*  1889*  1890   1892   1914*  1930*
```

```
                    1931    1933    1938    1939*   1946*   1947    1949*   1965*   1966    1968    2000*   2004*   2015*
                    2041*   2048*   2065*   2072*   2089*   2096*   2529*   2542*   2603*   2642*   2680*   2681*   2682
                    2710*   2711*   2712    2752*   2769*   2784*   2809*   2818*   2825    3226*   3257*   3258*   3260
                    3276*   3280    3290*   3318*   3369*   3476*   3481*   3499*
XMTINT  001264       704#   1044*   1240*   2109*   2128*   2136*   2521*
XMTLVL  001266       705#   2135
XMTTST  004106      1214#   2152    2163    2174    2185    2196    2207    2218    2229    2240    2251    2262    2273
                    2284    2295    2306    2317
Y       = 000020     533#   2144    2154#   2155    2165#   2166    2176#   2177    2187#   2188    2198#   2199    2209#
                    2210    2220#   2221    2231#   2232    2242#   2243    2253#   2254    2264#   2265    2275#   2276
                    2286#   2287    2297#   2298    2308#   2309    2319#   2320#   2322    2332#   2333    2343#   2344
                    2354#   2355    2365#   2366    2376#   2377    2387#   2388    2398#   2399    2409#   2410    2420#
                    2421    2431#   2432    2442#   2443    2453#   2454    2464#   2465    2475#   2476    2486#   2487
                    2497#   2836#   2837    2847#   2848    2858#   2859    2869#   2870    2880#   2891    2891#   2892
                    2902#   2903    2913#   2914    2924#   2925    2935#   2936    2946#   2947    2957#   2958    2968#
                    2969    2979#   2980    2990#   2991    3001#   3002    3012#   3014#   3015    3025#   3026    3036#
                    3037    3047#   3048    3058#   3059    3069#   3070    3080#   3081    3091#   3092    3102#   3103
                    3113#   3114    3124#   3125    3135#   3136    3146#   3147    3157#   3158    3168#   3169    3179#
                    3180    3190#
SCTLW   017135      1189    3680#
SENDAD  002500       670     913#
SSWREG  017106      3657    3675#
SVALUE  017115      3660    3663    3677#
.       = 017731     536#    537     539     541     543     545     547     553     555     557     559     561     563
                     565     567     569     573     575     577     579     581     583     585     587     589     591
                     593     595     597     599     601     603     605     607     609     611     613     615     617
                     619     621     623     625     627     629     631     633     635     637     639     641     643
                     645     647     649     651     653     655     657     659     661     663     665     669#    671#
                     675#    679#    686#    692#    694#    696#    713#    715#    800     803     837     1002    1119
                    1222    1287    1332    1334    1363    1366    1436    1457    1538    1541    1570    2598    2635
                    2638    2674    2676    2705    2707    2828    2830    3311    3469    3484    3563    3620#   3622#
.BAR    001624       755#   2070*
.BASRE  001630       757#
.BKCSR  001626       756#   2094*
.CSR    001622       754#   1068    2046*
```

DZDMAB.SRC CROSS REFERENCE TABLE -- MACRO NAMES

```
ALLMEN    424#  2656
BASBIN    424#  1920
BASRAN    424#  1952
BRKBIN    424#  1845
BRKRAN    424#  1877
BYTE      424#  2028  2052  2076
BYTREG    424#  2028  2052  2076
COMMEN      1#
ENDCOM      1#
ESCAPE      1#
GET       424#   849  1028  1049  1076  3579  3583  3664
GETPRI      1#
GETSWR      1#
HALFDU    424#  3450
HEADER    424#  1652  1669  1688  1707  1726  1745  1764  1783  1802  1821  1845  1877  1903  1920
         1952   1979  2006  2028  2052  2076  2100  2118  2499  2556  2582  2613  2656  2739  2791
         3191   3293  3378  3390  3402  3414  3426  3438  3450  3486
MULBRK    424#  3293
MULT        1#   424#  3191
NEWTST      1#
NOGO      424#  2582
POP         1#
PRIORI    424#  2118
PUSH        1#
REF       424#  1652
REFRBA    424#  1652
REPORT      1#
RESETD    424#  3486
RSTBAR    424#  2006
RSTBRK    424#  1903
RSTCSR    424#  1821
RUNBRK    424#  2837  2848  2859  2870  2881  2892  2903  2914  2925  2936  2947  2958  2969  2980
         2991   3002
RWB       424#  1669  1688  1707  1726  1745  1764  1783  1802
SETPRI      1#
SETUP       1#
SKIP        1#
SLASH       1#
STARS       1#
SWRSU       1#
TEST      424#  3015  3026  3037  3048  3059  3070  3081  3092  3103  3114  3125  3136  3147  3158
         3169   3180
TOTST1    424#  2498
TOTST2    424#  2556
TSTA      424#  3015  3026  3037  3048  3059  3070  3081  3092  3103  3114  3125  3136  3147  3158
         3169   3180
TSTAA     424#  3015  3026  3037  3048  3059  3070  3081  3092  3103  3114  3125  3136  3147  3158
         3169   3180
TSTBAR    424#  1979
TSTBB     424#  2144  2155  2166  2177  2188  2199  2210  2221  2232  2243  2254  2265  2276  2287
         2298   2309
TSTC      424#  2322  2333  2344  2355  2366  2377  2388  2399  2410  2421  2432  2443  2454  2465
         2476   2487
TSTCA     424#  2613
TSTCC     424#  2322  2333  2344  2355  2366  2377  2388  2399  2410  2421  2432  2443  2454  2465
         2476   2487
```

DZDMAB.SRC CROSS REFERENCE TABLE -- MACRO NAMES

```
TSTD      424#  2144  2155  2166  2177  2188  2199  2210  2221  2232  2243  2254  2265  2276  2287
          2298  2309
TSTF      424#  2837  2848  2859  2870  2881  2892  2903  2914  2925  2936  2947  2958  2969  2980
          2991  3002
TSTFF     424#  2837  2848  2859  2870  2881  2892  2903  2914  2925  2936  2947  2958  2969  2980
          2991  3002
TSTINT    424#  2100
TSTPT2    424#  2791
TSTRCV    424#  2322  2333  2344  2355  2366  2377  2388  2399  2410  2421  2432  2443  2454  2465
          2476  2487
TSTRWB    424#  1669  1688  1707  1726  1745  1764  1783  1802
TSTXMT    424#  2144  2155  2166  2177  2188  2199  2210  2221  2232  2243  2254  2265  2276  2287
          2298  2309
TYPBIN      1#
TYPDEC      1#
TYPNAM      1#
TYPNUM      1#
TYPOCS      1#
TYPOCT      1#
TYPTXT      1#
XMTDLY    424#  3378  3390  3402  3414  3426  3438
XMT100    424#  2739
$$ESCA      1#
$$NEWT      1#
$$SKIP      1#
.EQUAT      1#
.HEADE      1#
.KT11       1#
.SETUP      1#
.SWRHI      1#
.SACTI      1#
.SAPTB      1#
.SAPTH      1#
.SAPTY      1#
.SASTA      1#
.SCATC      1#
.SCMTA      1#
.SDB2D      1#
.SDB2O      1#
.SDIV       1#
.SEOP       1#
.SERRO      1#
.SERRT      1#
.SMULT      1#
.SPOWE      1#
.SRAND      1#
.SRDDE      1#
.SRDOC      1#
.SREAD      1#
.SR2AZ      1#
.SSAVE      1#
.SSB2D      1#
.SSB2O      1#
.SSCOP      1#
.SSIZE      1#
.SSUPR      1#
```

```
        3506  3519
EMT      515   516   517   518   519   520   521   522   523   524   525   526   527   528
HALT     538   540   542   544   546   548   554   556   558   560   562   564   566   568   570
         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   806   867  1195
INC     1070  1292  1458  1874  2625  2697  2758  2817  2818
INCB    1949
JMP      680   681   683   856   862   877   883   899  1018  1201  1535  1648  1651
JSR      768   780   784   788   813   817   844   849   874   886   913   998  1005  1007  1025
        1028  1049  1076  1215  1217  1283  1329  1391  1398  1444  1499  1513  1519  1525  1551
        1577  1599  1888  1963  2152  2163  2174  2185  2196  2207  2218  2229  2240  2251  2262
        2273  2284  2295  2306  2317  2330  2341  2352  2363  2374  2385  2396  2407  2418  2429
        2440  2451  2462  2473  2484  2495  2512  2595  2753  2845  2856  2867  2878  2889  2900
        2911  2922  2933  2944  2955  2966  2977  2988  2999  3010  3023  3034  3045  3056  3067
        3078  3089  3100  3111  3122  3133  3144  3155  3166  3177  3188  3207  3216  3304  3388
        3400  3412  3424  3436  3448  3537  3538  3554  3555  3579  3583  3598  3604  3610  3658
        3664
MOV      775   776   778   779   801   831   833   835   838   840   845   853   854   855   857
         858   859   861   870   871   879   896   910   917   918   919   920   921   922   927
         929   930   934   937   938   939   940   941   942   943   944   945   951   952   953
         954   955   956   957   959   960   966   968   973   977   983   984   989   991  1014
        1015  1016  1017  1032  1042  1044  1059  1060  1066  1067  1068  1069  1080  1090  1091
        1104  1105  1106  1107  1108  1109  1111  1115  1122  1133  1134  1135  1153  1155  1177
        1184  1194  1199  1214  1219  1225  1231  1232  1235  1236  1240  1244  1248  1278  1281
        1282  1297  1301  1304  1311  1313  1324  1336  1341  1343  1360  1361  1389  1395  1397
        1400  1401  1409  1410  1415  1421  1422  1428  1430  1432  1443  1454  1460  1461  1467
        1468  1469  1471  1486  1487  1488  1497  1498  1500  1501  1502  1504  1510  1511  1512
        1517  1523  1546  1554  1558  1559  1565  1567  1568  1571  1573  1574  1575  1576  1588
        1598  1633  1635  1640  1646  1649  1661  1663  1678  1697  1716  1735  1754  1773  1792
        1811  1831  1834  1838  1840  1856  1857  1863  1866  1871  1886  1887  1889  1890  1891
        1898  1913  1916  1931  1932  1938  1941  1946  1961  1962  1965  1966  1967  1974  1988
        1989  1990  2000  2001  2018  2024  2037  2041  2042  2045  2048  2049  2061  2065  2066
        2069  2072  2073  2085  2089  2090  2093  2096  2097  2109  2110  2127  2128  2129  2130
        2135  2136  2138  2514  2515  2516  2517  2520  2521  2524  2527  2541  2542  2548  2552
        2565  2566  2568  2577  2580  2602  2603  2609  2624  2626  2627  2628  2629  2631  2632
        2633  2642  2643  2653  2669  2670  2671  2672  2694  2700  2701  2702  2703  2717  2750
        2752  2769  2774  2783  2802  2803  2807  2808  2811  2813  3211  3213  3214  3215  3221
        3224  3239  3240  3253  3254  3257  3268  3269  3277  3278  3279  3302  3308  3309  3317
        3318  3323  3324  3333  3334  3343  3344  3354  3355  3362  3363  3364  3369  3370  3463
        3464  3465  3466  3467  3475  3476  3479  3495  3496  3497  3498  3500  3501  3502  3503
        3509  3512  3516  3521  3525  3529  3556  3564  3602  3603  3608  3614  3615  3625  3626
        3627  3632  3633  3634  3635  3668
MOVB     992  1000  1004  1006  1158  1160  1380  1474  1489  1514  1543  1582  1584  1597  2679
        2680  2699  2709  2710  3259  3543  3570  3597  3644
NEG     3609
NOP      872   890   914   915   916  1024  1092  1141  1243  1300  1462  1595  2113  2131  2139
        2498  2507  2508  2540  2693  3534  3551
RESET    912   969  1198  2668
ROL      974   975   978   979   981   982  1637  2651
ROR     1123  1125  1476  1477  1478  1613  1614  1615  1616  1617  1618  1619
RORB    1590
RTI      770   808   946   961   970   995  1438  1668  3636  3649  3669
RTS      763   821   923   985  1003  1093  1127  1142  1185  1448  1463  1482  1493  1505  1622
        1641  3593  3616
```

M07

DZDMAB.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
SEC      1547  1636  2650
SUB       928  1088  1433  1638  1665
TRAP      529
TST       804   834  1030  1078  1183  1228  1250  1253  1286  1296  1331  1369  1540  1566  1569
         1601  1605  1662  1684  1703  1722  1741  1760  1779  1798  1817  1995  2567  2597  2648
         2673  2689  2698  2704  2761  2777  2781  2819  2827  2831  3218  3243  3247  3264  3288
         3325  3483  3539  3557  3565
TSTB      799   802  1001  1156  1175  1178  1181  1290  1307  1333  1362  1365  1377  1529  1537
         2600  2630  2634  2637  2675  2706  2755  2814  2829  3310  3345  3349  3468  3559
.ASCII   3672  3675  3677  3680  3681  3684  3687  3689  3692  3694  3696  3697  3701  3707  3711
         3718  3721  3725  3730  3735  3739  3740  3742  3745
.BYTE    3744
.ENABL      1   422
.END     3756
.LIST       1   421   424
.MACR     424
.MACRO      1
.NLIST      1   420   424
.REM        4
.REPT    2144  2322  2837  3015
.TITLE    419


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,NOW.SEQ/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DZDMAB.SRC
RUN-TIME: 36 49 5 SECONDS
RUN-TIME RATIO: 232/92=2.5
CORE USED:  41K  (81 PAGES)
```

Spooler runtime 14 Seconds, 62 KCS, 390 disk reads, 3 disk writes, 90 pages