REM %

IDENTIFICATION

PRODUCT CODE: AC-8536D-MC

PRODUCT NAME: CZDMADO DM11A LGC TST

DATE RELEASED. APRIL 1978

MAINTAINER· DIAGNOSTIC GROUP

1. ABSTRACT

TWO SEPARATE DIAGNOSTIC PROGRAMS ARE PROVIDED FOR TESTING THE DM11A (ASYNCHRONOUS DATA MULTIPLEXER), CZDMA (DM11A LOGIC TESTS), AND CZDMB (DM11A MULTIPLE LINE DATA TESTS). THE LOGIC TESTS INDIVIDUALLY TEST EACH OF THE 16 DM11 LINES AND ALL COMMON 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 CONTROL 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? ;;++D
RESPONSE: CHARACTER LENGTH REFERS TO THE NUMBER OF DATA BITS PER CHARACTER (5-8). CARRIAGE RETURN DEFAULTS A CHAR-LENGTH OF "8". IF A CHARACTER LENGTH 5-7 IS DESIRED, TYPE THE VALUE(5-7) OF THE DESIRED LENGTH AT THE KEYBOARD WHEN PROMPTED.

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.

5.0 PROGRAM DESCRIPTION

5 1 PRGO - LOGIC TESTS

PRGO CONSISTS OF 152(8) INDEPENDENT ROUTINES WHICH TEST VARIOUS FUNCTIONS OF THE DM11 HARDWARE ANY OF THESE ROUTINES MAY BE INDIVIDUALLY SELECTED AND RUN (SEE SEC 4.2 FOR SWITCH SETTING)

5.1.1 ROUTINE DESCRIPTION

ROUTINE	TESTS
RT0	TESTS THE ABILITY TO REFERENCE THE FOUR DM11 REGISTERS CONTROL STATUS REGISTER (CSR), BUFFER ACTIVE REGISTER (BAR), BREAK STATUS REGISTER (BKCSR), AND THE BASE REGISTER (BASREG) IF AN ILLEGAL REFERENCE OCCURS WHEN THE CSR IS REFERENCED THE PROGRAM WILL INDICATE AN ERROR, AND AUTOMATICALLY LOOP THE ERROR AS LONG AS THE ERROR CONDITION EXISTS. RT0 PC=XXXXXX
RT1-RT10	BIT 'BANGS' THE CSR (BITS 0,1,2,4,5,6,12,13), TESTING THAT EACH BIT IN THE CSR CAN BE INDIVIDUALLY SET AND CLEARED TWO ERROR TYPES ARE DETECTED IN THESE TESTS, A BIT FAILED TO SET, AND/OR A BIT FAILED TO CLEAR. THE ERROR PRINTOUT SHOWS THE ROUTINE THAT FAILED AND THE PC WHERE THE ERROR WAS DETECTED.
RT11	TESTS THAT RESET AND CLEAR CLEAR ALL R/W BITS IN THE CSR TWO ERROR TYPES ARE DETECTED IN THIS ROUTINE SHOWING THE CONTENTS OF THE CSR AFTER THE RESET & CLEAR INSTRUCTION. THE PROGRAM AUTOMATICALLY LOOPS IF AN ERROR OCCURS. SHOWN BELOW IS THE ERROR TYPEOUT 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 NUM-
BER 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 REG-
ISTER (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 TRANSMITTED 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 AND 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 TRANSMITTING 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 TRANSMITTING FROM WHEN ERROR OCCURRED, FOR ERRORS IN OTHER 4K BLOCKS THE CORRECT RESULT CORRELATES TO THE ADDRESS WHERE THE ERROR OCCURRED. 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 CONTINUOUSLY 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 CORRECT 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 80 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  CZDMADO DM11A LGC TST
                                                LIST   ME,BIN,SEQ
                                               .ENABLE ABS,AMA

                                        ;CZDMADO DM11A LGC TST
                                        ;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
177776                                          CC=177776
177776                                          PSW=177776
000004                                          ERRVEC=4                ,ADDRESS OF ERROR TRAP VECTOR
000240                                          NOP=240
000000                                          OPEN=0
100000                                          MANUAL=BIT15
100000                                          LBIT17=100000
040000                                          LBIT16=40000
020000                                          LBIT15=20000
010000                                          LBIT14=10000
004000                                          LBIT13=4000
002000                                          LBIT12=2000
001000                                          LBIT11=1000
000400                                          LBIT10=400
000200                                          LBIT7=200
000100                                          LBIT6=100
000040                                          LBIT5=40
000020                                          LBIT4=20
000010                                          LBIT3=10
000004                                          LBIT2=4
000002                                          LBIT1=2
000001                                          LBIT0=1
100000                                          BIT15=100000
040000                                          BIT14=40000
020000                                          BIT13=20000
010000                                          BIT12=10000
004000                                          BIT11=4000
002000                                          BIT10=2000
001000                                          BIT9=1000
000400                                          BIT8=400
000200                                          BIT7=200
000100                                          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
```

```
  522            177777                          RTLAST=-1
  523
  524            000000                          Y=0
  525            177777                          X=-1
  526            000000                          A=0
  527            000000                          .=0
  528  000000   000002                          .+2                     ,UNASSIGNED TRAP
  529  000002   000000                          HALT
  530  000004   000006                MACHER:   .+2                     ;SP OVERFLOW, BUS ERROR TRAP
  531  000006   000000                          HALT
  532  000010   000012                          .+2                     ;RESERVED INSTRUCTION TRAP
  533  000012   000000                          HALT
  534  000014   000016                          .+2                     ;TRACE TRAP
  535  000016   000000                          HALT
  536  000020   000022                          .+2                     ;TRAP TO CALL IOX
  537  000022   000000                          HALT
  538  000024   000026                          .+2                     ;POWER FAIL TRAP
  539  000026   000000                          HALT
  540  000030   003100                          EMTINT                  ,EMT TRAP
  541  000032   000340                          PRTY7
  542  000034   006000                          DLY                     ;TRAP TRAP. SIMILAR TO EMT
  543  000036   000340                          PRTY7
  544  000040   000042                          .+2
  545  000042   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
  546  000044   000046                          .+2
  547  000046   000000                          HALT                    ,TRAPPED TO PREVIOUS ADDRESS
  548  000050   000052                          .+2
  549  000052   000000                          HALT                    ,TRAPPED TO PREVIOUS ADDRESS
  550  000054   000056                          .+2
  551  000056   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
  552  000060   000062                          .+2
  553  000062   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS
  554  000064   000066                          .+2
  555  000066   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  556  000070   000072                          .+2
  557  000072   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  558  000074   000076                          .+2
  559  000076   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  560  000100   000102                          .+2
  561  000102   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  562  000104   000106                          .+2
  563  000106   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  564  000110   000112                          .+2
  565  000112   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  566  000114   000116                          .+2
  567  000116   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  568  000120   000122                          .+2
  569  000122   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  570  000124   000126                          .+2
  571  000126   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  572  000130   000132                          .+2
  573  000132   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  574  000134   000136                          .+2
  575  000136   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
  576  000140   000142                          .+2
  577  000142   000000                          HALT                    ;TRAPPED TO PREVIOUS ADDRESS.
```

```
  578  000144  000146                .+2
  579  000146  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  580  000150  000152                .+2
  581  000152  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS
  582  000154  000156                .+2
  583  000156  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  584  000160  000162                .+2
  585  000162  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS
  586  000164  000166                .+2
  587  000166  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  588  000170  000172                .+2
  589  000172  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS
  590  000174  000176                .+2
  591  000176  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  592  000200  000202                 +2
  593  000202  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  594  000204  000206                 +2
  595  000206  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  596  000210  000212                .+2
  597  000212  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  598  000214  000216                 +2
  599  000216  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS
  600  000220  000222                .+2
  601  000222  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  602  000224  000226                 +2
  603  000226  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS
  604  000230  000232                .+2
  605  000232  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS
  606  000234  000236                .+2
  607  000236  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  608  000240  000242                .+2
  609  000242  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  610  000244  000246                .+2
  611  000246  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  612  000250  000252                .+2
  613  000252  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  614  000254  000256                .+2
  615  000256  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  616  000260  000262                .+2
  617  000262  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  618  000264  000266                .+2
  619  000266  000000                HALT                ,TRAPPED TO PREVIOUS ADDRESS.
  620  000270  000272                .+2
  621  000272  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  622  000274  000276                .+2
  623  000276  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  624  000300  000302                .+2
  625  000302  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  626  000304  000306                .+2
  627  000306  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  628  000310  000312                .+2
  629  000312  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  630  000314  000316                .+2
  631  000316  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
  632  000320  000322                .+2
  633  000322  000000                HALT                ;TRAPPED TO PREVIOUS ADDRESS.
```

```
634  000324  000326          +2
635  000326  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
636  000330  000332          +2
637  000332  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
638  000334  000336          +2
639  000336  000000          HALT          ;TRAPPED TO PREVIOUS ADDRESS
640  000340  000342          +2
641  000342  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
642  000344  000346          +2
643  000346  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
644  000350  000352          +2
645  000352  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
646  000354  000356          +2
647  000356  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
648  000360  000362          +2
649  000362  000300          HALT          ,TRAPPED TO PREVIOUS ADDRESS
650  000364  000366          +2
651  000366  000000          HALT          ,TRAPPED TO PPEVIOUS ADDRESS
652  000370  000372          +2
653  000372  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
654  000374  000376          +2
655  000376  000000          HALT          ,TRAPPED TO PREVIOUS ADDRESS
656
```

```
                    000046                          =46
000046  003036                          $ENDAD
                    000052                          .=52
000052  060000                          60000

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

                    000200                          .=200
000200  000137  002360                  JMP     @#START         ;GO TO START OF DIAGNOSTIC
000204  000137  002426                  JMP     @#RSTAT1                ;GO GET PROGRAM # & RESTART PROGRAM
                                                                ;USING PREVIOUS DM11 PARAMETERS
000210  000137  002514                  JMP     @#RSTAT2                ;RESTART PREVIOUS PROGRAM USING
                                                                ;PREVIOUS DM11 PARAMETERS

                    001200                          .=1200                                  ;;++D
001200  000000              SPBOT:      0
001202  177570              SWR         177570
001204  175570              DISPLAY:    175570
                    001400                          .=1400                                  ;;++D
001400  000000              CAT:        OPEN                ;STARTING ADDRESS OF
                    001440                          .=CAT+32            ;CURRENT ADDRESS TABLE
001440  000000              WCT:        OPEN                ;STARTING ADDRES OF
                    001500                          .=WCT+32            ;WORD COUNT TABLE
001500  000000              BAT         OPEN                ;STARTING ADDRESS OF
                    001540                          .=BAT+32            ;BIT ASSEMBLY TABLE
001540  000000              VAC.        OPEN                ;32. SPARE WORDS
001542  175000              CSR:        175000              ;ADDRESS OF CLOCK STATUS REGISTER
001544  175002              BAR:        175002              ;ADDRESS OF BUFFER ACTIVE REGISTER
001546  175004              BKCSR:      175004              ;ADDRESS OF BREAK STATUS REGISTER
001560  175006              BASREG:     175006              ;ADDRESS OF BASE REGISTER
001562  000000              CLKINT:     OPEN                ;DM11 VECTOR ADDRESS (RECEIVER)
001564  000240              CLKLVL:     PRTY5               ;PRIORITY LEVEL
001566  000000              XMTINT:     OPEN                ;DM11 VECTOR ADDRESS (TRANSMITTER)
001560  000240              XMTLVL:     PRTY5               ;TRANSMITTER PRIORITY LEVEL
001562  000000              TTDAT:      OPEN                ;TUMBLE TABLE DATA
001564  000000              LINBIT:     OPEN                ;LINE BIT (FOR BAR)
001566  000000              RCVDAT:     OPEN
001570  000000              XMTDAT:     OPEN
001572  000000              CARMSK:     OPEN
                    001600                          .=VAC+32
001600  000000              TUMTAB:     OPEN                ;STARTING ADDRESS OF
                    002000                          .=TUMTAB+128.       ;TUMBLE TABLE
002000  000000              KSTART:     OPEN                ;CURRENT PROGRAM START ADDRESS
002002  000000              CURTST:     OPEN                ;CONTAINS ADDR OF CURRENT TEST
002004  000000              RTNNO:      OPEN                ;CONTAINS CURRENT TEST #
002006  000000              NXTST:      OPEN                ;CONTAINS ADDR OF NEXT TEST
002010  000000              ICTR:       OPEN                ;CONTAINS CURRENT ITERATION COUNT
002012  000000              SCOPTR:     OPEN                ;CONTAINS CURRENT SCOPE POINTER
002014  000000              PRGLIM:     OPEN
002016  007076              PRGTAB:     PRG0                ;PRG0 START ADDRESS
002020  016514                          PRG1                ;PRG1 START ADDRESS
002022  016574                          PRG2                ;PRG2 START ADDRESS
```

```
  713  002024  007120                RSTART: PRGOR                   ;PRGO RESTART ADDRESS
  714  002026  016524                        PRG1R                   ;PRG1     "       "
  715  002030  016604                        PRG2R                   ;PRG2     "       "
  716  002032  003322                EMTTAB: TYP                     ;POINTER TO TYPEOUT ROUTINE
  717  002034  002146                        ERR                     ;POINTER TO ERROR ROUTINE
  718  002036  002126                        DTCHK                   ;POINTER TO DATA COMPARISON ROUTINE
  719  002040  000000                        0
  720  002042  000000                        0
  721  002044  003232                        SRSETT                  ;POINTER TO RESET ROUTINE
  722  002046  002700                        ESCOPE                  ;POINTER TO SCOPE ROUTINE
  723  002050  003132                        SAVRG                   ;POINTER TO SAVE REGISTERS ROUTINE
  724  002052  003172                        RSTRG                   ;POINTER TO RESTORE REGISTERS ROUTINE
  725  002054  002164                        ERR1                    ;POINTER TO ERROR1 ROUTINE
  726  002056  003434                        INIT                    ;POINTER TO INITIALIZE ROUTINE
  727  002060  017412                        SUSWRR
  728  002062  017472                        KBDINTT
  729  002064  017546                        CNTLUU
  730  002066  177560                TKCSR:  177560
  731  002070  177562                TKDBR:  177562
  732  002072  177564                TPCSR:  177564
  733  002074  177566                TPDBR:  177566
  734  002076  000000                COUNT:  OPEN
  735  002100  000000                PCADD:  OPEN
  736  002102  000000                APCADD: OPEN
  737  002104  000000                PRVCNT: OPEN
  738  002106  175001                . CSR:  175001
  739  002110  175003                . BAR:  175003
  740  002112  175005                  BKCSR: 175005
  741  002114  175007                  BASREG: 175007
  742  002116  000000                PASS:   OPEN
```

```
                                        ;ROUTINE TO TYPE OUT INCORRECT ROUTINE SELECTED
002120  104000                          INCRTN. TYPE
002122  020017                                  M1                      ;TYPE INCORRECT ROUTINE SELECTED
002124  000207                                  RTS     %7              ;EXIT.

                                        ;DATA COMPARISON ROUTINE.
002126  123737  001566  001570  DTCHK:  CMPB    RCVDAT,XMTDAT   ;COMPARE RECEIVED & TRANSMITTED DATA
002134  001403                          BEQ     1$              ;CHARS. BRANCH IF SAME
002136  004737  002332                  JSR     7,CNVDAT        ;CONVERT RCVDAT & XMTDAT TO ASCII
002142  104011                          ERROR1
002144  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:
002146  012737  000402  002246  ERR:    MOV     #402,ERRB       ;MOV BR .+6 TO ERRB
002154  013737  002100  002102          MOV     PCADD,APCADD    ;GET PC WHERE ERROR OCCURRED
002162  000410                          BR      ERRA
002164  012737  000240  002246  ERR1:   MOV     #240,ERRB       ;MOVE NOP TO ERRB
002172  013737  002100  002102          MOV     PCADD,APCADD    ;GET PC WHERE ERROR OCCURRED
002200  004737  002332                  JSR     7,CNVDAT        ;CONVERT RCVDAT & XMIT DAT TO ASCII
002204  104014                  ERRA:   KBDIN                   ;GO CHECK FOR  G
002206  032777  020000  176766          BIT     #BIT13,@SWR     ;ERROR PRINTOUT DESIRED
002214  001017                          BNE     ERRC            ;BRANCH IF NO PRINTOUT
002216  004537  006154                  JSR     5,OACNV         ;CONVERT
002222  002102                          APCADD                  ;DATA
002224  020224                          APC                     ;TO
002226  000006                          6                       ;ASCII
002230  004537  006154                  JSR     5,OACNV         ;FOR
002234  002004                          RTNNO                   ;PRINTOUT
002236  020216                          ATNUMB
002240  000003                          3
002242  104000                          TYPE                    ;TYPE ERROR
002244  020213                          EMO                     ;MESSAGE
002246  000000                  ERRB:   OPEN                    ;NOP IF ERROR1, BR  +6 IF ERROR
002250  104000                          TYPE                    ;TYPE ANOTHER MESSAGE
002252  017754                          ERDAT                   ;IF ERROR 1
002254  032777  000400  176720  ERRC:   BIT     #BIT8,@SWR      ;RING BELL ON ERROR?
002262  001411                          BEQ     ERRD            ;BRANCH IF NO BELL ON ERROR
002264  105777  177602                  TSTB    @TPCSR          ;TELEPRINTER
002270  100375                          BPL     .-4             ;READY?
002272  012777  000007  177574          MOV     #BELL,@TPDBR    ;RING THE BELL
002300  105777  177566                  TSTB    @TPCSR          ;WAIT FOR THE BELL TO RING
002304  100375                          BPL     .-4
002306  023737  000042  000046  ERRD:   CMP     @#42,@#46       ;ACT11?
002314  001403                          BEQ     ERRHLT
002316  005777  176660                  TST     @SWR            ;HALT ON ERROR
002322  100001                          BPL     ERREX           ;GO TO EXIT IF NO HALT ON ERROR
002324  000000                  ERRHLT: HALT
002326  104014                  ERREX:  KBDIN                   ;CHECK FOR   G
002330  000002                          RTI                     ;RETURN


                                        ;SUBROUTINE TO CONVERT RCVDAT AND XMTDAT TO ASCII AND PLACE
```

```
 799                                    ;IN MESSAGE
 800  002332  004537  006154             CNVDAT: JSR     5,OACNV
 801  002336  001570                             XMTDAT
 802  002340  017766                             AASB
 803  002342  000006                             6
 804  002344  004537  006154                     JSR     5,OACNV
 805  002350  001566                             RCVDAT
 806  002352  020002                             AWAS
 807  002354  000006                             6
 808  002356  000207                             RTS     7               ;EXIT
 809
 810
```

```
                                ;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.

002360  012706  001200          START:  MOV     #SPBOT,%6
002364  104013                          SUSWR                   ;SEE IF SWITCH-LESS PROCESSOR
002366  012737  002410  000004          MOV     #1$,@#ERRVEC    ;SET UP FOR ERROR TRAP
002374  005737  172300                  TST     @#172300        ;TEST FOR KT11
002400  012737  012716  012330          MOV     #RT70,@#RT65+2  ;KT11 PRESENT, SET UP TO
                                                                ;SKIP RT66 AND RT67
002406  000402                          BR       +6
002410  012706  001200          1$:     MOV     #SPBOT,%6       ;TRAP OCCURRED, NO KT11 PRESENT,
                                                                ;RESET STACK
002414  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC      ;RESET ERROR TRAP


                                                                ;OUT INTERRUPTS (SET PRIORITY LEVEL 7)
002422  004737  003464                  JSR     7,@#DMPAR               ;GET DM11 PARAMETERS
002426  012706  001200          RSTAT1: MOV     #SPBOT,%6
002432  104012                          INITIALIZE
002434  023737  000042  000046          CMP     @#42,@#46       ;ACT11?
002442  001405                          BEQ     PRGNUM+2
002444  104000                          TYPE
002446  020011                          MO
002450  004537  004416                  JSR     5,RECD          ;GET PRGNUM AND PUT IT
002454  000000          PRGNUM: 0       ;HERE
002456  043737  002014  002454          BIC     PRGLIM,PRGNUM   ;MASK OFF UNUSED BITS
002464  006337  002454                  ASL     PRGNUM          ;SHIFT PROGRAM #
002470  012737  004576  000024          MOV     #PFAIL,24
002476  012737  000340  000026          MOV     #PRTY7,26
002504  013700  002454                  MOV     PRGNUM,%0       ;GET PROGRAM #
002510  000170  002016                  JMP     @PRGTAB(0)      ;GO START PROGRAM
002514  012737  004576  000024  RSTAT2. MOV     #PFAIL,24
002522  012737  000340  000026          MOV     #PRTY7,26
002530  012706  001200                  MOV     #SPBOT,%6
002534  104012                          INITIALIZE
002536  013700  002454                  MOV     PRGNUM,%0       ;GET PROGRAM #
002542  000170  002024                  JMP     @RSTART(0)      ;GO RESTART PROGRAM
002546  022737  000176  001202  SRSET:  CMP     #SWREG,SWR
002554  001410                          BEQ     1$
002556  023737  000042  000046          CMP     @#42,@#46       ;ACT11?
002564  001406                          BEQ     GETRDY
002566  104000                          TYPE                    ;TYPE MESSAGE TO REQUEST SWITCH
002570  020030                          MJ                      ;REGISTER SETTINGS
002572  000000                          HALT                    ;WAIT FOR OPERATOR TO SET SWITCHES
002574  000401                          BR      GETRDY
002576  104015          1$:     CNTLU                           ;GO GET SWREG SETTINGS
002600  013737  002000  002006  GETRDY: MOV     KSTART,NXTST    ;ADDR OF 1ST ROUTINE TO NXTST
002606  012737  000006  000004  GTRDYX: MOV     #6,@#ERRVEC             ;RESET ERROR TRAP VECTOR
002614  104012                          INITIALIZE
002616  004737  003046          GTRDYA. JSR     %7,FORWD        ;ROLL FORWARD TO "NEXT" ROUTINE.
002622  032777  001000  176352          BIT     #BIT9,@SWR      ;CHECK SELECT ROUTINE SWITCH
002630  001003                          BNE     GTRDYC          ;BRANCH IF SELECT ROUTINE SWITCH IS SET
```

```
                                        JMP     @CURTST         ;GO RUN CURRENT ROUTINE
                                        BR      SCOPED          ;NO GO. MANUAL RTN BYPASSED
                        GTRDYC  MOV     @SWR,%0         ;(SR) TO R0
                                        BIC     #177600,%0      ;MASK UNDESIRED BITS
                                        CMPB    RTNNO,%0        ;COMPARE RTNNO TO (R0)
                                        BNE     GTRDYD          ;BRANCH IF ROUTINE NOT FOUND YET
                                        JMP     @CURTST         ;GO RUN ROUTINE.
                        GTRDYD. CMP     #-1,NXTST       ;NO.  CHECK FOR LAST ROUTINE
                                        BNE     GTRDYA          ;BRANCH IF NOT LAST ROUTINE
                                        JSR     %7,INCRTN       ;YES.  INCORRECT ROUTINE SELECTED
                                        BR      GETRDY          ;START OVER

                        ;SCOPE SERVICE ROUTINE
                        ESCOPE. NOP
                                        KBDIN                   ;CHECK FOR  G
                                        CLR     @#AR            ;CLEAR ALL DM11 REGISTERS
                                        CLR     @#CSR           ;AND SET BASE REGISTER
                                        CLR     @#KCSR          ;AT THE STARTING ADDRESS
                                        INITIALIZE
                                        MOV     SCOPTR,(SP)
                                        BIT     #BIT14,@SWR     ;CHECK FOR SCOPE OPTION
                                        BEQ     SCOPEB          ;BRANCH IF SCOPE SW NOT SET
                        SCOPEA: JMP     @(SP)           ;RETURN TO ROUTINE
                        SCOPEB: BIT     #BIT11,@SWR     ;TEST INHIBIT ITERATION SWITCH
                                        BNE     SCOPED          ;BRANCH IF INHIBIT ITERATION SW SET
                                        CMP     @#42,@#46       ;ACT11?
                                        BNE     1$              ;BR IF NO
                                        TST     @#PASS          ;1ST PASS?
                                        BEQ     SCOPED          ;BR IF YES
                        1$:     DEC     ICTR            ;DECREMENT ITERATION COUNT
                                        BNE     SCOPEA          ;BRANCH IF COUNT NOT 0
                        SCOPED: BIT     #BIT9,@SWR      ;CHECK SELECT ROUTINE SWITCH
                                        BNE     GETRDY          ;BRANCH IF SELECT RTN SW SET
                                        CMP     #-1,NXTST       ;LAST TEST?
                                        BNE     GTRDYX          ;BRANCH IF NOT LAST TEST
                                        INC     @#PASS
                                        TYPE                    ;TYPE
                                        M2                      ;'PRGEND'
                                        MOV     @#42,%0         ;CHECK XXDP/ACT11 MONITOR HOOK
                                        BEQ     GETRDY
                                        RESET
                        SENDAD: JSR     7,(0)           ;RETURN TO XXDP/ACT11 MONITOR
                                        NOP
                                        NOP
                                        NOP
                        FORWD:  MOV     NXTST,%5        ;ADDR OF NEXT ROUTINE TO R5
                                        MOV     (5)+,RTNNO      ;GET NEXT ROUTINE NUMBER.
                                        MOV     (5)+,NXTST      ;GET ADDR OF NEXT "NEXT" ROUTINE
                                        MOV     (5)+,ICTR       ;GET ITERATION COUNT.
                                        MOV     (5)+,SCOPTR     ;GET SCOPE LOOP ENTRY POINTER.
                        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
```

```
 923  003102  162716  000002                SUB     #2,(6)          ,FORM PC OF EMT INSTRUCTION
 924  003106  011637  002100                MOV     (6),PCADD       ,GET PC OF EMT INSTRUCTION
 925  003112  017616  000000                MOV     @(6),(6)        ,GET EMT INSTRUCTION
 926  003116  105066  000001                CLRB    1(6)            ,CLEAR MSH OF EMT INSTRUCTION
 927  003122  006316                        ASL     (6)             ,SHIFT EMT IDENTIFIER
 928  003124  062716  002032                ADD     #EMTTAB,(6)
 929  003130  013607                        MOV     @(6)+,X7        ,GO TO PROPER EMT
 930
 931                                  ,SAVE REGS 0 TO 4 SUBROUTINE
 932  003132  012637  003166          SAVRG:  MOV     (6)+,SVRPC      ,SAVE PC AND PSW
 933  003136  012637  003170                  MOV     (6)+,SVRPSW
 934  003142  010446                        MOV     X4,-(6)         ,SAVE REGS 0 - 4
 935  003144  010346                        MOV     X3,-(6)         ,IN STACK
 936  003146  010246                        MOV     X2,-(6)
 937  003150  010146                        MOV     X1,-(6)
 938  003152  010046                        MOV     X0,-(6)
 939  003154  013746  003170                MOV     SVRPSW,-(6)     ,RESTORE PC AND PSW
 940  003160  013746  003166                MOV     SVRPC,-(6)
 941  003164  000002                        RTI                     ,EXIT
 942  003166  000000                  SVRPC:  OPEN
 943  003170  000000                  SVRPSW: OPEN
 944
 945                                  ;RESTORE REGS 0 TO 4 SUBROUTINE.
 946  003172  012637  003226          RSTRG:  MOV     (6)+,RSTPC      ,SAVE PC AND PSW.
 947  003176  012637  003230                  MOV     (6)+,RSTPSW
 948  003202  012600                        MOV     (6)+,X0         ;RESTORE REGS 0 - 4
 949  003204  012601                        MOV     (6)+,X1         ,FROM STACK
 950  003206  012602                        MOV     (6)+,X2
 951  003210  012603                        MOV     (6)+,X3
 952  003212  012604                        MOV     (6)+,X4
 953
 954  003214  013746  003230                MOV     RSTPSW,-(6)     ,RESTORE PC AND PSW
 955  003220  013746  003226                MOV     RSTPC,-(6)
 956  003224  000002                        RTI                     ;EXIT
 957  003226  000000                  RSTPC:  OPEN
 958  003230  000000                  RSTPSW: OPEN
 959
 960                                  ;ROUTINE TO ISSUE RESET.
 961  003232  012700  052525          SRSETT: MOV     #52525,X0               ;DATA TO R0.
 962  003236  005100                        COM     X0              ;COMPLEMENT (R0).
 963  003240  010037  003234                MOV     X0,SRSETT+2     ;(R0) TO SRSETT+2.
 964  003244  000005                        RESET                   ;ISSUE RESET.  (R0) IS
 965  003246  000002                        RTI                     ;DISPLAYED.  EXIT.
 966
 967                                  ;RANDOM NUMBER GENERATOR.  ROUTINE EXITS WITH NUMBER IN REGISTER 0.
 968  003250  013700  003316          RNGEN:  MOV     RP1,X0
 969  003254  006100                  RP'     X0
 970  003256  006100                        ROL     X0
 971  003260  063700  003320                ADD     RP2,X0
 972  003264  010037  003316                MOV     X0,RP1
 973  003270  006100                        ROL     X0
 974  003272  006100                        ROL     X0
 975  003274  063700  003320                ADD     RP2,X0
 976  003300  006100                        ROL     X0
 977  003302  006100                        ROL     X0
 978  003304  010037  003320                MOV     X0,RP2
```

```
  979  003310  013700  003316                MOV     RP1,%0
  980  003314  000207                      RTS     %7              ,EXIT  NUMBER IN R0
  981  003316  001233              RP1     1233
  982  003320  007622              RP2.    7622
  983                               ,SUBROUTINE TO OUTPUT ASCII MESSAGE ON TELETYPE PRINTER
  984  003322  011600              TYP     MOV     @%6,%0          ;GET ADDRESS THAT CONTAINS MESSAGE ADDRESS
  985  003324  062716  000002              ADD     #2,@%6          ;SET UP EXIT.
  986  003330  011000                      MOV     @%0,%0          ;ADDRESS OF MESSAGE TO R0.
  987  003332  112037  003432      TYPA:   MOVB    (0)+,TYPDAT     ;GET CHARACTER
  988  003336  122737  000100  003432      CMPB    #100,TYPDAT     ;CHECK FOR"@"CHARACTER
  989  003344  001001                      BNE     TYPC            ,BRANCH IF NOT"@".
  990  003346  000002                      RTI                     ;TERMINATOR CHAR. DONE  EXIT
  991  003350  122737  000045  003432  TYPC  CMPB    #45,TYPDAT      ,CHECK FOR"%".
  992  003356  001412                      BEQ     TYPF            ,BRANCH IF"%"
  993  003360  004737  003366              JSR     %7,TYPD         ;TYPE CHAR IN TYPDAT
  994  003364  000762                      BR      TYPA
  995  003366  113777  003432  176500  TYPD  MOVB    TYPDAT,@TPDBR   ;OUTPUT CHARACTER TO PRINTER
  996  003374  105777  176472              TSTB    @TPCSR          ;WAIT FOR DONE FLAG.
  997  003400  100375                      BPL     -4
  998  003402  000207                      RTS     %7              ,EXIT
  999  003404  112737  000015  003432  TYPF  MOVB    #15,TYPDAT      ,MOVE CARRIAGE RETURN CODE TO TYPDAT
 1000  003412  004737  003366              JSR     %7,TYPD         ;GO TYPE CHAR.
 1001  003416  112737  000012  003432  TYPG. MOVB    #12,TYPDAT      ;MOVE LF CODE TO TYPDAT
 1002  003424  004737  003366              JSR     %7,TYPD         ;GO TYPE CHAR.
 1003  003430  000740                      BR      TYPA
 1004  003432  000000              TYPDAT: OPEN
 1005
 1006
 1007                               ;SUBROUTINE TO INITIALIZE STACK POINTER AND SET PROCESSOR PRIORITY
 1008                               ,LEVEL 7
 1009  003434  012777  001400  176106  INIT.   MOV     #CAT,@BASREG    ;INITIALIZE THE BASE REGISTER
 1010  003442  012737  000340  177776          MOV     #PRTY7,PSW              ;SET PRIORITY LEVEL 7
 1011  003450  011637  001200              MOV     (SP),SPBOT      ;GET RETURN ADDRESS
 1012  003454  012706  001200              MOV     #SPBOT,SP       ;SET BOTTOM OF THE STACK
 1013  003460  000176  000000              JMP     @(SP)           ;RETURN
 1014
 1015
 1016                               ;SUBROUTINE TO GET DM11 PARAMETERS
 1017                               ;VECTOR ADDRESS
 1018  003464  023737  000042  000046  DMPAR:  CMP     @#42,@#46       ;ACT11?
 1019  003472  001060                      BNE     6$              ;BR IF NO
 1020                                       ;SIZE FOR INTERRUPT VECTOR IN AUTO MODE
 1021  003474  012700  000302              MOV     #302,R0         ;SET UP FLOATING VECT AREA
 1022  003500  010060  177776      4$:     MOV     R0,-2(R0)
 1023  003504  012720  000003              MOV     #3,(R0)+
 1024  003510  005720                      TST     (R0)+
 1025  003512  022700  000776              CMP     #776,R0
 1026  003516  100370                      BPL     4$
 1027  003520  012737  003610  000014          MOV     #5$,@#14        ;SET BPT VECT
 1028  003526  012737  000340  000016          MOV     #340,@#16       ;& PSW
 1029  003534  012737  177777  001440  3$:     MOV     #-1,WCT         ;SET TO XMIT 1 CHAR
 1030  003542  012737  017102  001400          MOV     #OUTBUF,CAT
 1031  003550  012777  000105  175764          MOV     #BIT6+BIT2+BIT0,@CSR    ;SET IE
 1032  003556  005037  177776                  CLR     @#PSW           ;LVL 0
 1033  003562  012777  000001  175754          MOV     #LBIT0,@BAR     ;XMIT
 1034  003570  012737  177777  002076          MOV     #-1,COUNT       ,WAIT
```

```
1035  003576  005337  002076          2$      DEC     COUNT
1036  003602  001375                          BNE     2$
1037  003604  104001                          ERROR                   ;NO INT OCCURRED
1038  003606  000752                          BR      3$              ;REPEAT IT
1039  003610  162716  000004          5$:     SUB     #4,(SP)         ;CALC INT VECT
1040  003614  011637  003650                  MOV     (SP),@#VECTOR   ;STORE IT
1041  003620  012737  000016  000014          MOV     #16,@#14        ;RESTORE BPT VECT
1042  003626  004737  004372                  JSR     7,OVRLAY        ;.+2, HALT IN VECT AREA
1043  003632  000415                          BR      VECOK
1044  003634  004737  004372          6$:     JSR     7,OVRLAY        ;PUT HALT,.+2 IN VECTOR AREA
1045  003640  104000                          TYPE                    ;ASK USER FOR RECEIVER INT  VECTOR
1046  003642  017626                          WHERE                   ;OF UNIT UNDER TEST
1047  003644  004537  004416                  JSR     5,RECD          ;GET VECTOR AND PUT IT
1048  003650  000000          VECTOR: 0                               ;HERE
1049  003652  005737  003650                  TST     VECTOR
1050  003656  001003                          BNE     VECOK
1051  003660  012737  000300  003650          MOV     #300,VECTOR     ;SET VECTOR = TO 0300
1052  003666  023727  003650  000300  VECOK:  CMP     VECTOR,#300     ;IS VECTOR HIGHER OR
1053  003674  103003                          BHIS    VECOKB          ;EQUAL TO 0300
1054  003676  104000          VECOKA: TYPE                            ;TYPE '?'
1055  003700  020017                          M1
1056  003702  000670                          BR      DMPAR           ;ASK FOR ANOTHER VECTOR
1057  003704  023727  003650  000770  VECOKB: CMP     VECTOR,#770     ;IS VECTOR = TO OR
1058  003712  101371                          BHI     VECOKA          ;LESS THAN 770
1059  003714  032737  000007  003650          BIT     #7,VECTOR       ;LSB OF VECTOR MUST BE ALL 0'S
1060  003722  001365                          BNE     VECOKA
1061  003724  013737  003650  001552          MOV     VECTOR,CLKINT
1062  003732  062737  000004  003650          ADD     #4,VECTOR
1063  003740  013737  003650  001556          MOV     VECTOR,XMTINT
1064
1065                          ;UNIT NUMBER
1066  003746  023737  000042  000046  DMPARB: CMP     @#42,@#46       ;ACT11?
1067  003754  001405                          BEQ     UNIT+2          ;BR IF YES
1068  003756  104000                          TYPE
1069  003760  017725                          WHICH
1070  003762  004537  004416                  JSR     5,RECD          ;GET UNIT AND PUT IT
1071  003766  000000          UNIT    0                               ;HERE
1072  003770  023727  003766  000017          CMP     UNIT,#17        ;UNIT SELECTED MUST BE
1073  003776  101403                          BLOS    UNTOKA          ;BETWEEN 0 & 17
1074  004000  104000                          TYPE
1075  004002  020017                          M1
1076  004004  000760                          BR      DMPARB
1077  004006  006337  003766          UNTOKA: ASL     UNIT
1078  004012  006337  003766                  ASL     UNIT
1079  004016  006337  003766                  ASL     UNIT
1080  004022  012702  000004                  MOV     #4,X2
1081  004026  012701  001542                  MOV     #CSR,X1
1082  004032  042711  000370          UNTOKB: BIC     #370,(1)        ;FORM ADDRESSES OF
1083  004036  063721  003766                  ADD     UNIT,(1)+       ;REGISTERS OF UNIT SELECTED
1084  004042  005302                          DEC     X2
1085  004044  001372                          BNE     UNTOKB
1086
1087  004046  012702  000004                  MOV     #4,X2
1088  004052  012703  001542                  MOV     #CSR,X3
1089  004056  012701  002106                  MOV     #.CSR,X1
1090  004062  012311          UNTOKC: MOV     (3)+,(1)                ;FORM ODD BYTE ADDRESSES
```

```
1091  004064  005221                          INC     (1)+
1092  004066  005302                          DEC     %2
1093  004070  001374                          BNE     UNTOKC
1094                                  ;CHARACTER LENGTH
1095  004072  023737  000042  000046  DMPARC: CMP     @#42,@#46       ;ACT11?
1096  004100  001405                          BEQ     LENGTH+2        ;BR IF YES
1097  004102  104000                          TYPE
1098  004104  017740                          LEVEL
1099  004106  004537  004416                  JSR     5,RFCD          ;GET LENGTH AND PUT IT
1100  004112  000000                  LENGTH: 0                       ;HERE
1101  004114  005737  004112                  TST     LENGTH
1102  004120  001003                          BNE     LENOKA
1103  004122  012737  000010  004112          MOV     #8.,LENGTH
1104  004130  023727  004112  000005  LENOKA: CMP     LENGTH,#5       ;CHARACTER LENGTH SELECTED MUST
1105  004136  103003                          BHIS    LENOKC          ;BE BETWEEN 5-8
1106  004140  104000                  LENOKB: TYPE                    ;CARRIAGE RETURN SELECTS 8
1107  004142  020017                          M1
1108  004144  000752                          BR      DMPARC
1109  004146  023727  004112  000010  LENOKC: CMP     LENGTH,#8.
1110  004154  101371                          BHI     LENOKB
1111  004156  162737  000005  004112          SUB     #5,LENGTH
1112  004164  006337  004112                  ASL     LENGTH
1113  004170  013701  004112                  MOV     LENGTH,%1
1114  004174  016137  004206  001572          MOV     LENOKD(1),@#CARMSK      ;SET CHARACTER LENGTH MASK
1115  004202  000240                          NOP
1116  004204  000207                          RTS     7               ;EXIT PARAMETERS ROUTINE
1117
1118                                  ;THE BELOW TABLE REPRESENTS THE CHARACTER LENGTH MASK FOR 5,6,7, AND 8
1119                                  ;BITS PER CHARACTER RESPECTIVELY.
1120  004206  177740                  LENOKD: 177740
1121  004210  177700                          177700
1122  004212  177600                          177600
1123  004214  177400                          177400
1124
1125                                  ;CALCULATE MACHINE TIME TO TRANSMIT ONE CHARACTER
1126  004216  005037  004366          TIMER:  CLR     TIME1
1127  004222  012737  177777  001440          MOV     #-1,WCT         ;SET UP TO TRANSMIT
1128  004230  012737  017102  001400          MOV     #OUTBUF,CAT     ;1 CHARACTER ON LINE 1
1129  004236  012777  004336  175306          MOV     #TIMEC,@CLKINT  ;LOAD RECEIVER INTERRUPT
1130  004244  012777  000340  175302          MOV     #PRTY7,@CLKLVL  ;AND PRIORITY
1131  004252  012777  000001  175264          MOV     #LBIT0,@BAR     ;START TRANSMITTING
1132  004260  012777  000105  175254          MOV     #BIT6+BIT2+BIT0,@CSR    ;SET IE BIT
1133  004266  006037  177776                  CLR     @#PSW           ;SET PROCESSER PRIORITY LEVEL = 0
1134  004272  012737  000044  002076  TIMEA:  MOV     #44,COUNT
1135  004300  062737  000001  004366          ADD     #1,TIME1        ;INCREMENT MACH. TIME TO TRANSMIT
1136  004306  001007                          BNE     TIMEB
1137  004310  006077  175226                  CLR     @CSR
1138  004314  012737  000340  177776          MOV     #PRTY7,PSW      ;SET PROCESSER PRIORITY LEVEL = 7
1139  004322  104001                          ERROR                   ;TRANSMITTER FAILED TO INTERRUPT
1140  004324  000734                          BR      TIMER
1141  004326  006337  002076          TIMEB:  DEC     COUNT
1142  004332  001375                          BNE     .-4
1143  004334  000756                          BR      TIMEA
1144  004336  006077  175200          TIMEC:  CLR     @CSR
1145  004342  013737  004366  004370          MOV     TIME1,TIME14
1146  004350  006037  004370                  ROR     TIME14
```

```
1147  004354  000241                        CLC
1148  004356  006037  004370                ROR     TIME14
1149  004362  022626                        POPSP2                  ;RESTORE STACK POINTER
1150  004364  000207                        RTS     7               ;EXIT TIME CALCULATION ROUTINE
1151
1152  004366  000000                TIME1.  OPEN                    ;CONTAINS MACHINE TIME TO XMIT 1 CHAR
1153  004370  000000                TIME14: OPEN                    ;CONTAIN TIME TO XMIT 1/4 CHAR
1154
1155                                ;SUBROUTINE TO PUT HALT,.+2 IN VECTOR AREA (0300-1000)
1156  004372  012702  000302        OVRLAY: MOV     #302,R2
1157  004376  010262  177776        1$:     MOV     R2,-2(R2)
1158  004402  005022                        CLR     (R2)+
1159  004404  005722                        TST     (R2)+
1160  004406  022702  000776                CMP     #776,R2
1161  004412  100371                        BPL     1$
1162  004414  000207                        RTS     7
1163
1164
1165                                ;SUBROUTINE TO RECEIVE DATA
1166                                ;THIS SUBROUTINE RECEIVES DATA FROM THE KEYBOARD (UP TO SIX OCTAL
1167                                ;DIGITS AND PLACES THEM INTO THE ADDRESS FOLLOWING THE SUBROUTINE
1168                                ;CALL (JSR 5,RECD).  NO REGISTER CONTENTS ARE DISTURBED.
1169
1170
1171                                ;SUBROUTINE TO INPUT DATA FROM TTY
1172
1173  004416  010046                RECD:   MOV     R0,-(SP)
1174  004420  005015                1$:     CLR     (5)             ;CLEAR OLD DATA
1175  004422  012737  000007  004574        MOV     #7,CNT          ;SET CHAR COUNT
1176  004430  105777  175432        2$:     TSTB    @TKCSR          ;WAIT FOR CHAR
1177  004434  100375                        BPL     2$
1178  004436  117700  175426                MOVB    @TKDBR,R0
1179  004442  142700  000200                BICB    #200,R0         ;STRIP OFF PARITY
1180  004446  110077  175422                MOVB    R0,@TPDBR       ;ECHO CHARACTER
1181  004452  122700  000025                CMPB    #25,R0          ;IS IT A  U
1182  004456  001443                        BEQ     5$              ;BRANCH IF YES
1183  004460  122700  000015                CMPB    #15,R0          ;IS IT A <CR>
1184  004464  001415                        BEQ     6$              ;BRANCH IF YESS
1185  004466  142700  000060                BICB    #60,R0
1186  004472  132700  000110                BITB    #110,R0         ;CHECK FOR 0-7 (8)
1187  004476  001031                        BNE     7$              ;BRANCH IF NOT
1188  004500  006315                        ASL     (5)
1189  004502  006315                        ASL     (5)
1190  004504  006315                        ASL     (5)             ;SHIFT DATA
1191  004506  150015                        BISB    R0,(5)          ;INSET NEW CHAR
1192  004510  005337  004574                DEC     CNT
1193  004514  001422                        BEQ     7$              ;ONLY 6 CHAR'S PLEASE
1194  004516  000744                        BR      2$              ;NEXT CHARACTER
1195  004520  105777  175346        6$:     TSTB    @TPCSR
1196  004524  100375                        BPL     6$              ;WAIT FOR READY
1197  004526  012777  000012  175340        MOV     #12,@TPDBR      ;TYPE <LF>
1198  004534  105777  175332        8$:     TSTB    @TPCSR
1199  004540  100375                        BPL     8$              ;WAIT FOR READY         ;,++D
1200  004542  005077  175326                CLR     @TPDBR          ;LOAD CHAR
1201  004546  105777  175320        9$:     TSTB    @TPCSR
1202  004552  100375                        BPL     9$
```

```
1203  004554  005725                          TST     (R5)+                   ;ADJUST R5
1204  004556  012600                          MOV     (SP)+,R0                ;RESTORE R0
1205  004560  000205                          RTS     R5
1206  004562  104000                  7$      TYPE
1207  004564  020017                          M1
1208  004566  104000                  5$:     TYPE
1209  004570  017666                          $CTLU
1210  004572  000712                          BR      1$                      ;START OVER
1211  004574  000000                  CNT:    0
1212
1213                                  ;POWER FAIL ROUTINE
1214  004576  012737  004606  000024  PFAIL:  MOV     @PWRUP,24
1215  004604  000000                          HALT
1216
1217                                  ;POWER UP SUBROUTINE
1218  004606  000005                  PWRUP:  RESET                           ;GIVE TELEPRINTER TIME TO START
1219  004610  012706  001200                  MOV     @SPBOT,%6
1220  004614  104001                          ERROR                           ;TYPE POWER FAIL ERROR
1221  004616  000137  002514                  JMP     @#RSTAT2                ;GO RESTART PROGRAM
1222
1223                                  ;LINE TEST SUBROUTINE: THIS LINE TEST PROVIDES SEVERAL TESTS ON A DM11 LINE
1224                                  ;THE SUBROUTINE IS CALLED BY JSR 5, LNTST.  THIS INSTRUCTION PROVIDES THE
1225                                  ;LINE BIT AND LINE NUMBER.  THE FOLLOWING LINE TESTS ARE PERFORMED
1226                                  ;WAITS UNTIL CHARACTER SHOULD HAVE BEEN TRANSMITTED, THEN TESTS
1227
1228                                          ;START BAR BIT CLEARED          ;DO NEXT TEST IF ERROR
1229                                          ;READY SET                      ;DO NEXT TEST IF ERROR
1230                                          ;WORD COUNT WENT TO 0           ;DO NEXT TEST IF ERROR
1231                                          ;CURRENT ADDRESS DID NOT INCREMENT ;DO NEXT TEST IF ERROR
1232                                          ;INTERRUPTS TO CORRECT VECTOR   ;DO NEXT TEST IF ERROR (NO INTERRUPT)
1233                                          ;READY BIT CAN BE CLEARED       ;END OF TEST
1234  004622  012537  016730          XMTTST: MOV     (5)+,@#LINE             ;GET LINE NUMBER
1235  004626  004737  007044                  JSR     7,GTLINB                ;GO FORM LINE BIT (FOR BAR)
1236  004632  005037  001570                  CLR     XMTDAT
1237  004636  004537  006246          1$:     JSR     5,@#XMITD               ;GO TO TRANSMIT SUBROUTINE
1238  004642  177777                          -1                              ;TRANSMIT ONE CHARACTER
1239  004644  012703  000010                  MOV     #10,%3                  ;WAIT IN
1240  004650  005002                          CLR     %2                      ;THIS
1241  004652  005302                  2$:     DEC     %2                      ;LOOP
1242  004654  001376                          BNE     .-2                     ;UNTIL THE
1243  004656  005303                          DEC     %3                      ;TRANSMITTER
1244  004660  001374                          BNE     2$                      ;IS FINISHED
1245  004662  017737  174656  001566          MOV     @#BAR,RCVDAT            ;BAR SHOULD NOW BE CLEAR
1246  004670  001401                          BEQ     3$                      ;BRANCH IF IT IS
1247  004672  104011                          ERROR1                          ;ERROR! BAR BIT FAILED TO CLEAR
1248  004674  005777  174642          3$:     TST     @#CSR                   ;TEST READY BIT SHOULD BE SET
1249  004700  100401                          BMI     4$                      ;BRANCH IF SET
1250  004702  104001                          ERROR                           ;ERROR! READY NOT SET
1251  004704  013701  016730          4$:     MOV     @#LINE,%1               ;GET LINE NUMBER
1252  004710  016137  001440  001566          MOV     WCT(1),RCVDAT           ;WORD COUNT SHOULD BE 0
1253  004716  001401                          BEQ     5$
1254  004720  104011                          ERROR1                          ;ERROR! WORD COUNT NOT EQUAL TO 0
1255  004722  012737  017102  001570  5$:     MOV     #OUTBUF,XMTDAT ;
1256  004730  016137  001400  001566          MOV     CAT(1),RCVDAT           ;CURRENT ADDRESS SHOULD NOT HAVE INCREMENTED
1257  004736  023737  001566  001570          CMP     RCVDAT,XMTDAT           ;
1258  004744  001401                          BEQ     6$
```

```
1259  00·746  104011                          ERROR1                  ;ERROR! CURRENT ADDR. DID NOT INCREMENT
1260  004750  012777  005002  174600  6$.     MOV     #7$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
1261  004756  052777  010000  174556          BIS     #BIT12,@CSR     ;ENABLE TRANSMITTER INTERRUPT
1262  004764  005037  177776                  CLR     @#PSW           ;SET PROCESSOR PRIORITY =0
1263  004770  000240                          NOP
1264  004772  012737  000340  177776          MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1265  005000  104001                          ERROR                   ;TRANSMITTER FAILED TO INTERRUPT OR
1266                                  ;INTERRUPTED TO WRONG LOCATION AND HALTED WITH ADDRESS +2 DISPLAYED
1267  005002  022626                  7$:     CMP     (6)+,(6)+       ;RESET STACK PTR
1268  005004  012737  000340  177776          MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1269  005012  042777  110000  174522          BIC     #BIT12+BIT15,@CSR       ;CLEAR XMIT IE & READY BITS
1270  005020  005777  174516                  TST     @CSR            ;TEST THAT READY CLEARED
1271  005024  100001                          BPL     8$              ;GO TO EXIT
1272  005026  104001                          ERROR                   ;ERROR! READY FAILED TO CLEAR
1273  005030  005726                  8$:     TST     (6)+            ;RESET STACK PTR
1274  005032  104006                          SCOPE                   ;SCOPE
```

```
1275
1276
1277                                    ,RECEIVER LINE TESTS
1278                                    ,THE RECEIVER LINE TEST SUBROUTINE IS ENTERED WITH
1279                                    ;A JSR 5, RCVTST INSTRUCTION FOLLOWED BY THE
1280                                    ;LINE BIT AND LINE NUMBER OF THE LINE TO BE
1281                                    ,TESTED.  THE SUBROUTINE PERFORMS THE FOLLOWING
1282                                    ;TEST AS SHOWN BELOW  IN THE EVENT OF AN ERROR
1283                                    ,THE REMAINING TESTS ARE ABORTED
1284                                    ,TEST SEQUENCE AND ADDRESS TAG
1285                                    ,
1286                                    ,       CHARACTER DONE SETS                             RTSTA
1287                                    ,       CHARACTER DONE CAUSES INTERRUPT                 RTSTB
1288                                    ,       CHARACTER DONE CAN BE CLEARED                   RTSTC
1289                                    ,       TUMBLE TABLE ENTRY IS CORRECT                   RTSTD
1290                                    ,       NO ENTRY IN NEXT TABLE ADDRESS                  RTSTE
1291                                    ,       HARDWARE TABLE POINTER INCREMENTED              RTSTF
1292                                    ,       NEXT ENTRY WAS CORRECT                          RTSTG
1293
1294                                    ,NOTES:  IF THE HARDWARE PROVIDES AN INCORRECT VECTOR
1295                                    ,        ADDRESS THE PROGRAM WILL HALT AND DISPLAY
1296                                    ;        THE INCORRECT VECTOR+2 IN THE ADDRESS LIGHTS.
1297
1298  005034  012737  177777  017102  RCVTST: MOV     #-1,OUTBUF      ;LOAD ALL 1'S INTO OUTPUT BUFFER
1299  005042  005037  001600                  CLR     TUMTAB          ;CLEAR THE FIRST
1300  005046  005037  001602                  CLR     TUMTAB+2        ;TWO TUMBLE TABLE ADDRESSES
1301  005052  012737  000340  177776          MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1302  005060  012537  016730                  MOV     (5)+,LINE       ;GET LINE NUMBER
1303  005064  004537  006246                  JSR     5,@#XMITD       ;TRANSMIT 1 CHARACTER (0'S)
1304  005070  177777                          -1                      ;ON LINE SPECIFIED BY JSR
1305  005072  052777  000001  174442          BIS     #BIT0,@CSR      ;SET GO BIT
1306  005100  005777  174436                  TST     @CSR            ;WAIT FOR TRANSMITTER
1307  005104  100375                          BPL     .-4             ;TO TRANSMIT 1 CHAR.
1308  005106  042777  100000  174426          BIC     #BIT15,@CSR     ;CLEAR TRANSMITTER READY FLAG
1309  005114  005046                          CLR     -(SP)           ;SET WATCH DOG TIMER
1310  005116  105777  174420          1$:     TSTB    @CSR            ;TEST CHAR. DONE FLAG
1311  005122  100404                          BMI     2$              ;BRANCH IF SET
1312  005124  005216                          INC     (SP)            ;WAIT FOR THE FLAG
1313  005126  001373                          BNE     1$
1314  005130  104001                          ERROR                   ;ERROR! CHAR. DONE FLAG FAILED TO SET
1315  005132  000660                          BR      8$              ;GO TO EXIT
1316  005134  005726                  2$:     TST     (SP)+           ;RESTORE STACK PTR
1317  005136  012777  005172  174406          MOV     #3$,@CLKINT     ;LOAD RECEIVER INTERRUPT VEC. ADRS.
1318  005144  052777  000100  174370          BIS     #BIT6,@CSR      ;SET RECEIVER IE BIT
1319  005152  005037  177776                  CLR     @#PSW           ;ENABLE INTERRUPTS
1320  005156  000240                          NOP
1321  005160  012737  000340  177776          MOV     #PRTY7,PSW      ;LOCK OUT INTERRUPTS
1322  005166  104001                          ERROR                   ;RECEIVER FAILED TO INTERRUPT
1323  005170  000631                          BR      8$              ;GO TO EXIT
1324  005172  012737  000340  177776  3$:     MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
1325  005200  022626                          CMP     (6)+,(6)+
1326  005202  042777  000300  174332          BIC     #BIT7+BIT6,@CSR ;CLEAR CHAR. DONE FLAG
1327  005210  105777  174326                  TSTB    @CSR            ;TEST THAT CHAR DONE FLAG CLEARED
1328  005214  100002                          BPL     4$              ;BRANCH IF CHAR. DONE FLAG CLEARED
1329  005216  104001                          ERROR                   ;ERROR! CHAR. DONE FAILED TO CLEAR
1330  005220  000515                          BR      8$              ;GO TO EXIT
```

```
1331  005222  013737  001600  001566  4$:     MOV     TUMTAB,RCVDAT   ;GET TUMBLE TABLE ENTRY
1332  005230  042737  020000  001566          BIC     #BIT13,RCVDAT   ;CLEAR PARITY INDICATOR
1333  005236  012737  000377  001570          MOV     #377,XMTDAT     ;LOAD XMTDAT WITH  TRANSMITTED DATA
1334  005244  043737  001572  001570          BIC     CARMSK,XMTDAT   ;CLEAR NON TRANSMITTED BITS
1335  005252  153737  016730  001571          BISB    LINE,XMTDAT+1   ;LOAD XMTDAT WITH PROPER LINE #
1336  005260  052737  100000  001570          BIS     #BIT15,XMTDAT   ;SET VALID DATA ENTRY BIT IN XMTDAT
1337  005266  023737  001566  001570          CMP     RCVDAT,XMTDAT   ;COMPARE TUMBLE TABLE ENTRY (RCVDAT) &
1338                                                                  ;CORRECT RESULT (XMTDAT)
1339
1340  005274  001402                          BEQ     5$
1341  005276  104011                          ERROR1                  ;ERROR! INCORRECT TUMBLE TABLE
1342  005300  000465                          BR      8$              ;ENTRY; GO TO EXIT
1343  005302  005037  001600          5$:     CLR     TUMTAB
1344  005306  013737  001602  001566          MOV     TUMTAB+2,RCVDAT ;GET NEXT ENTRY
1345  005314  001404                          BEQ     6$              ;BRANCH IF ALL 0'S
1346  005316  005037  001570                  CLR     XMTDAT
1347  005322  104011                          ERROR1                  ;ERROR! FALSE ENTRY IN NEXT
1348  005324  000453                          BR      8$              ;TUMBLE TABLE ADDRESS
1349  005326  004537  006246          6$:     JSR     5,@#XMITD       ;TRANSMIT 1 CHARACTER (ALL 1'S)
1350  005332  177777                          -1                      ;ON LINE SPECIFIED BY JSR
1351  005334  005777  174202                  TST     @CSR            ;WAIT FOR TRANSMITTER
1352  005340  100375                          BPL     .-4             ;READY FLAG
1353  005342  105777  174174                  TSTB    @CSR            ;TEST FOR THE DONE FLAG
1354  005346  100375                          BPL     .-4
1355  005350  042777  000200  174164          BIC     #BIT7,@CSR      ;CLEAR CHAR. DONE FLAG
1356  005356  013737  001600  001566          MOV     TUMTAB,RCVDAT   ;TEST THAT HARDWARE TUMBLE
1357  005364  001404                          BEQ     7$              ;TABLE POINTER INCREMENTED (+2)
1358  005366  005037  001570                  CLR     XMTDAT
1359  005372  104011                          ERROR1                  ;ERROR! TUMBLE TABLE POINTER DID
1360  005374  000427                          BR      8$              ;NOT INCREMENT; GO TO EXIT
1361  005376  013737  001602  001566  7$:     MOV     TUMTAB+2,RCVDAT ;GET TUMBLE TABLE ENTRY
1362  006404  042737  020000  001566          BIC     #BIT13,RCVDAT   ;CLEAR PARITY INDICATOR
1363  006412  012737  000377  001570          MOV     #377,XMTDAT     ;LOAD XMTDAT WITH TRANSMITTED DATA
1364  006420  043737  001572  001570          BIC     CARMSK,XMTDAT   ;CLEAR NON-TRANSMITTED BITS
1365  006426  153737  016730  001571          BISB    LINE,XMTDAT+1   ;LOAD LINE # INTO XMTDAT
1366  006434  052737  100000  001570          BIS     #BIT15,XMTDAT   ;SET VALID DATA ENTRY BIT INTO XMTDAT
1367  006442  023737  001566  001570          CMP     RCVDAT,XMTDAT   ;COMPARE TUMBLE TABLE ENTRY (RCVDAT) &
1368                                                                  ;CORRECT RESULT (XMTDAT)
1369  005450  001401                          BEQ     8$
1370  005452  104011                          ERROR1                  ;ERROR! 2ND TUMBLE TABLE ENTRY
1371  005454  104006                  8$:     SCOPE                   ;WAS INCORRECT; SCOPE
1372
1373                                  ;SUBROUTINE TO TEST BREAK OPERATION
1374                                  ;THE TRANSMITTER WILL TRANSMIT THE BREAK FOR TWO CHARACTER
1375                                  ;TIMES AND THEN THE FOLLOWING TESTS WILL BE PERFORMED
1376                                  ;       A VALID DATA ENTRY WAS MADE     BKTSTB
1377                                  ;       BREAK BIT SET                   BKTSTC
1378                                  ;       DATA WAS ALL 0'S                        BKTSTD
1379
1380  005456  012777  000001  174056  BRKTST: MOV     #1,@CSR         ;SET THE GO BIT
1381  005464  011577  174056                  MOV     (5),@#KCSR      ;SET THE BREAK BIT
1382  005470  105777  174046                  TSTB    @CSR            ;WAIT FOR THE RECEIVER TO
1383  005474  100375                          BPL     .-4             ;RECEIVE BREAK
1384  005476  042777  000200  174036          BIC     #BIT7,@CSR      ;CLEAR FLAG
1385  005504  105777  174032                  TSTB    @CSR            ;WAIT FOR THE RECEIVER TO
1386  005510  100375                          BPL     .-4             ;TO RECEIVE BREAK
```

```
1387  005512  042777  000200  174022          BIC     #BIT7,@CSR      ;CLEAR FLAG
1388  005520  005077  174022                  CLR     @#KCSR          ;CLEAR BREAK BIT
1389  005524  005737  001600          1$.     TST     TUMTAB          ;TEST FOR VALID DATA ENTRY
1390  005530  100402                          BMI     2$
1391  005532  104001                          ERROR                   ;ERROR! NO VALID DATA ENTRY
1392  005534  000421                          BR      4$              ;GO TO EXIT
1393  005536  032737  040000  001600  2$:     BIT     #BIT14,TUMTAB   ;TEST THAT BREAK BIT IS SET
1394  005544  001002                          BNE     3$              ;IN TUMBLE TABLE
1395  005546  104001                          ERROR                   ;ERROR! BREAK BIT FAILED TO SET
1396  005550  000413                          BR      4$              ;GO TO EXIT
1397  005552  105737  001600          3$      TSTB    TUMTAB          ;TEST THAT DATA IS ALL 0'S
1398  005556  001410                          BEQ     4$
1399  005560  005037  001566                  CLR     RCVDAT
1400  005564  113737  001600  001566          MOVB    TUMTAB,RCVDAT   ;GET RECEIVED DATA
1401  005572  005037  001570                  CLR     XMTDAT
1402  005576  104011                          ERROR1                  ;ERROR! DATA WAS NOT ALL 0'S
1403  005600  104006                  4$      SCOPE                   ;SCOPE
1404
1405
1406                                          ;SUBROUTINE TO TRANSMIT & RECEIVE ON ALL LINES THE DELAY BETWEEN
1407                                          ;TRANSMITTING ON A LINE IS SUPPLIED BY THE CALLING JSR INSTRUCTION
1408                                          ;NOTE NO DATA CHECKING IS PERFORMED BY THIS TEST
1409  005602  012537  002076          DLYXMT: MOV     (5)+,@#COUNT    ;GET CHARACTER DELAY COUNT
1410  005606  005037  001570                  CLR     XMTDAT
1411  005612  004537  006224                  JSR     5,BMOVE         ;LOAD OUTPUT BUFFER WITH DATA
1412  005616  020233                          MSG1                    ;TO BE TRANSMITTED
1413  005620  017102                          OUTBUF
1414  005622  000100                          64.
1415  005624  012737  000001  001564          MOV     #LBIT0,@#LINBIT
1416  005632  005037  016730                  CLR     @#LINE
1417  005636  012777  000001  173676  1$:     MOV     #BIT0,@CSR      ;SET THE GO BIT
1418  005644  004537  006246          2$:     JSR     5,@#XMITD       ;TRANSMIT 64. CHAR.
1419  005650  177700                          -64.                    ;ON A LINE
1420  005652  013737  004366  005666          MOV     @#TIME1,4$
1421  005660  013704  002076                  MOV     @#COUNT,%4      ;GET CHARACTER DELAY COUNT
1422  005664  104400                  3$:     DELAY
1423  005666  000000                  4$:     0
1424  005670  005304                          DEC     %4
1425  005672  001374                          BNE     3$
1426  005674  052737  000002  016730          ADD     #2,LINE         ;FORM NEXT LINE NUMBER
1427  005702  006337  001564                  ASL     LINBIT          ;SHIFT LINE BIT
1428  005706  103356                          BCC     2$              ;BRANCH IF ALL LINES NOT DONE
1429  005710  012704  000100                  MOV     #64.,%4
1430  005714  013737  004366  005724          MOV     TIME1,6$
1431  005722  104400                  5$:     DELAY
1432  005724  000000                  6$:     0
1433  005726  005304                          DEC     %4
1434  005730  001374                          BNE     5$
1435  005732  017737  173606  001566          MOV     @BAR,RCVDAT     ;GET & TEST BAR DATA
1436  005740  001402                          BEQ     7$              ;EXIT IF DONE
1437  005742  104011                          ERROR1                  ;ERROR! BAR SHOULD'VE BEEN CLEAR
1438  005744  000413                          BR      8$
1439  005746  022777  100201  173566  7$:     CMP     #100201,@CSR    ;TEST THAT ONLY DONE,GO,& READY BITS ARE SET
1440  005754  001407                          BEQ     8$
1441  005756  012737  100201  001570          MOV     #100201,XMTDAT
1442  005764  017737  173552  001566          MOV     @CSR,RCVDAT     ;GET CSR CONTENTS
```

```
1443  005772  104011                        ERROR1                  ;INCORRECT CSR COMTENTS
1444  005774  005726              8$:       POPSP                   ;RESET THE STACK
1445  005776  104006                        SCOPE                   ;SCOPE
1446
1447                                        ;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS
1448  006000  011637  006050      DLY       MOV     (6),3$          ;GET DELAY COUNT ADDRESS
1449  006004  062716  000002                ADD     #2,(6)          ;SET UP EXIT ADDRESS
1450  006010  017737  000034  006050       MOV     @3$,3$          ;GET DELAY COUNT
1451  006016  001413                        BEQ     2$              ;EXIT IF NO DELAY
1452  006020  012737  000050  006052 1$     MOV     #50,4$
1453  006026  162737  000001  006050       SUB     #1,3$
1454  006034  001404                        BEQ     2$
1455  006036  005337  006052               DEC     4$
1456  006042  001375                        BNE      -4
1457  006044  000765                        BR      1$
1458  006046  000002              2$        RTI                     ;EXIT
1459  006050  000000              3$        OPEN                    ;CONTAINS DELAY COUNT
1460  006052  000000              4$        OPEN                    ;CONTAINS DELAY ROUTINE CONSTANT
1461
1462                                        ;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS
1463  006054  012737  177777  006076 INBIN: MOV     #-1,RIND        ;SET ALL VARIABLES
1464  006062  004537  006224               JSR     %5,BMOVE        ;TO MINUS 1.
1465  006066  006076                        RIND
1466  006070  006077                        RIND+1
1467  006072  000005                        5
1468  006074  000207                        RTS     %7              ;EXIT
1469  006076  000000              RIND:     OPEN
1470  006100  000000              PTO.      OPEN
1471  006102  000000              PT1:      OPEN
1472
1473                                        ;SPECIAL BINARY COUNT PATTERN SUBROUTINE.  EXITS WITH BIN CHAR IN R1
1474  006104  013737  006100  006102 GTBIN: MOV     PTO,PT1         ;PREVIOUS BIN CHAR TO PT1
1475  006112  005137  006102               COM     PT1
1476  006116  005137  006076               COM     RIND
1477  006122  001002                        BNE     .+6
1478  006124  005237  006102               INC     PT1
1479  006130  042737  177400  006102       BIC     #177400,PT1     ;MASK TO 8 BITS
1480  006136  013737  006102  006100       MOV     PT1,PTO         ;SAVE BIN CHAR IN PTO
1481  006144  013701  006102               MOV     PT1,%1          ;BIN CHAR TO R1
1482  006150  000240                        NOP
1483  006152  000207                        RTS     %7              ;EXIT
1484
1485                                        ;OCTAL TO ASCII CONVERT ROUTINE
1486  006154  104007              OACNV:    SAVREG                  ;SAVE REGISTERS ON THE STACK
1487  006156  013504                        MOV     @(5)+,%4        ;GET OCTAL VALUE.
1488  006160  012501                        MOV     (5)+,%1         ;GET DESTINATION ADDR
1489  006162  012502                        MOV     (5)+,%2         ;GET CONVERT COUNT.
1490  006164  060201                        ADD     %2,%1           ;DEVELOP ADDR TO STORE 1ST CHAR
1491  006166  010403              OACNVA:   MOV     %4,%3
1492  006170  042703  177770               BIC     #177770,%3      ;ISOLATE LEAST SIGNIFICANT DIGIT
1493  006174  062703  000060               ADD     #60,%3          ;CONVERT DIGIT TO ASCII.
1494  006200  110341                        MOVB    %3,-(1)         ;STORE ASCII CHARACTER.
1495  006202  042704  000007               BIC     #7,%4
1496  006206  006004                        ROR     %4
1497  006210  006004                        ROR     %4
1498  006212  006004                        ROR     %4
```

```
1499  006214  005302                        DEC     %2              ,DONE ALL DIGITS?
1500  006216  001363                        BNE     OACNVA          ;BRANCH IF NOT DONE.
1501  006220  104010                        RSTREG                  ,RESTORE THE REGISTERS
1502  006222  000205                        RTS     %5              ;DONE.  EXIT.
1503
1504                                  ,SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
1505  006224  104307                  BMOVE:  SAVREG                  ;SAVE REGS.
1506  006226  012501                        MOV     (5)+,%1         ;GET"FROM"ADDRESS
1507  006230  012502                        MOV     (5)+,%2         ;GET"TO"ADDRESS
1508  006232  012503                        MOV     (5)+,%3         ;GET COUNT
1509  006234  112122                  1$:   MOVB    (1)+,(2)+       ;MOVE BYTE
1510  006236  005303                        DEC     %3              ;DECREMENT COUNT
1511  006240  001375                        BNE     1$              ,BRANCH IF NOT DONE
1512  006242  104010                        RSTREG                  ;RESTORE REGS.
1513  006244  000205                        RTS     %5              ;DONE EXIT
1514
1515                                  ,SUBROUTINE TO TRANSMIT DATA.  SUBROUTINE CALLED BY
1516                                  ,JSR 5,XMITD
1517  006246  010046                  XMITD:  MOV     %0,-(SP)        ,SAVE R0 ON THE STACK
1518  006250  013700  016730                 MOV     @#LINE,%0       ;GET LINE
1519  006254  004737  007044                 JSR     7,@#GTLINB      ,FORM LINE BIT (FOR BAR)
1520  006260  012777  001400  173262         MOV     #CAT,@#BASREG   ;INITIALIZE BASE REGISTER
1521  006266  012760  017102  001400         MOV     #OUTBUF,CAT(0)  ;LOAD FIRST CHAR ADDRESS IN CAT
1522  006274  012560  001440                 MOV     (5)+,WCT(0)     ,GET WORD COUNT
1523  006300  053777  001564  173236         BIS     @#LINBIT,@#BAR  ;LOAD LINE POSITION INTO BAR
1524  006306  012600                        MOV     (SP)+,%0        ;RESTORE R0
1525  006310  000205                        RTS     5               ,EXIT
1526
1527                                  ;ROUTINE TO TEST A LINE
1528                                  ;THE LINE TO BE TESTED IS PROVIDED BY THE JSR CALL TO THE ROUTINE
1529                                  ;100. CHARACTERS ARE TRANSMITTED,RECEIVED AND CHECKED BY THIS ROUTINE
1530  006312  012537  016730           DATTST: MOV     (5)+,@#LINE     ,GET LINE NUMBER
1531  006316  012737  000144  002076   DAT1AA: MOV     #100.,COUNT     ;GET CHARACTER COUNT
1532  006324  012702  017102                 MOV     #OUTBUF,%2      ;GET ADDRESS OF OUTPUT BUFFER
1533  006330  004737  006104           1$.   JSR     7,@#GTBIN       ;GET DATA
1534  006334  110122                        MOVB    %1,(2)+         ;LOAD OUTPUT BUFFER WITH DATA
1535  006336  005337  002076                 DEC     COUNT           ;GOT ALL DATA?
1536  006342  001372                        BNE     1$
1537  006344  012701  001600                 MOV     #TUMTAB,%1      ;LOAD TUMBLE TABLE POINTER
1538  006350  005037  001600                 CLR     TUMTAB
1539  006354  004537  006224                 JSR     5,BMOVE         ;CLEAR
1540  006360  001600                        TUMTAB                  ;TUMBLE
1541  006362  001601                        TUMTAB+1                ;TABLE
1542  006364  000177                        177                     ;64 WORDS
1543  006366  012702  017246                 MOV     #INBUF,%2       ;SETUP INPUT BUFFER POINTER
1544  006372  052777  000001  173142         BIS     #BIT0,@#CSR     ;SET THE GO BIT
1545  006400  004537  006246                 JSR     5,@#XMITD       ;TRANSMIT
1546  006404  177634                        -100.                   ,100. CHARACTERS
1547  006406  032777  160000  173126   2$:   BIT     #BIT15+BIT14+BIT13,@#CSR;TEST IF READY OR ANY ERROR
1548  006414  001004                        BNE     3$              ;FLAGS ARE SET
1549  006416  105777  173120                 TSTB    @#CSR           ;WAIT FOR THE RECEIVER
1550  006422  100371                        BPL     2$              ,TO RECEIVE A CHARACTER
1551  006424  000415                        BR      5$
1552  006426  032777  060000  173106   3$:   BIT     #BIT14+BIT13,@#CSR;TEST FOR ERROR FLAGS
1553  006434  001403                        BEQ     4$              ,BRANCH NO ERROR
1554  006436  104001                        ERROR
```

```
006440  000137  006760                   JMP     15$              ,GO EXIT
006444  042777  100000  173070   4$      BIC     #BIT15,@CSR      ,CLEAR TRANSMITTER READY FLAG
006452  105777  173064                   TSTB    @CSR             ,TEST FOR CHARACTER READY
006456  100375                           BPL     -4
006460  042777  000200  173054   5$      BIC     #BIT7,@CSR       ,CLEAR CHAR DONE BIT
006466  005711                           TST     (1)
006470  100401                           BMI     +4               ,TEST FOR VALID ENTRY
006472  104001                           ERROR                    ,REPORT INVALID ENTRY
006474  111122                           MOVB    (1),(2)+         ,MOVE CHAR FROM TUM. TAB  TO INPUT BUFFER

                                 ,ROUTINE TO STORE RECEIVED PARITY BIT IN PARITY BIT BUFFER
006476  012705  000001                   MOV     #1,%5            ;GET ROTATE COUNT
006502  000261                           SEC                      ,SET THE CARRY BIT
006504  032711  020000                   BIT     #BIT13,(1)       ,TEST RECEIVED PARITY BIT
006510  001001                           BNE     6$               ,BRANCH IF RECEIVED PARITY WAS ODD
006512  000241                           CLC                      ;CLEAR CARRY BIT
006514  004537  006764           6$      JSR     5,RORPARBUF      ,ROTATE RECEIVED PARITY INTO PARITY BUFFER

                                 ,ROUTINE TO TEST THAT ENTRY IS FOR THE CORRECT LINE
006520  011137  001562                   MOV     (1),@#TTDAT      ,GET TABLE ENTRY
006524  042737  160777  001562           BIC     #160777,TTDAT    ,CLEAR ALL BUT LINE NUMBER
006532  123737  016730  001563           CMPB    LINE,TTDAT+1     ;COMPARE LINE NUMBERS
006540  001410                           BEQ     7$
006542  013737  016730  001570           MOV     LINE,XMTDAT      ,GET CORRECT LINE # (X2)
006550  013737  001562  001566           MOV     TTDAT,RCVDAT     ,GET LINE # (X2) THAT FALSE DATA CAME IN ON
006556  104011                           ERROR1                   ,ERROR! DATA CAME IN ON A LINE THAT
                                                                  ,PROGRAM WAS NOT TRANSMITTING ON
006560  000477                           BR      15$              ,EXIT TEST
006562  020127  001776           7$      CMP     %1,#TUMTAB+176   ,IS POINTER AT THE END
006566  001002                           BNE     8$               ,OF THE TABLE
006570  012701  001576                   MOV     #TUMTAB-2,%1
006574  005721                   8$      TST     (1)+             ,INCREMENT POINTER
006576  010046                           MOV     %0,-(6)          ,SAVE REGISTER ZERO
006600  013700  016730                   MOV     LINE,%0          ;FETCH LINE NUMBER
006604  005760  001440                   TST     WCT(0)           ,HAS THE LAST CHARACTER BEEN TRANSMITTED
006610  001402                           BEQ     .+6              ;LAST CHARACTER HAS BEEN TRANSMITTED
006612  012600                           MOV     (6)+,%0          ;RESTORE REGISTER ZERO
006614  000674                           BR      2$               ,GO WAIT FOR NEXT CHARACTER
006616  012600                           MOV     (6)+,%0          ,RESTORE REGISTER ZERO
006620  012701  017102           9$      MOV     #OUTBUF,%1
006624  012702  017246                   MOV     #INBUF,%2
006630  012705  000014                   MOV     #12.,%5          ;ROTATE PARITY BUFFER
006634  004537  006764                   JSR     5,RORPARBUF      ;12. PLACES RIGHT
006640  005037  001566           10$     CLR     RCVDAT
006644  005037  001570                   CLR     XMTDAT
006650  020127  017245                   CMP     %1,#OUTBUF+99    ;HAVE ALL CHARS  BEEN COMPARED
006654  001441                           BEQ     15$
006656  112137  001570                   MOVB    (1)+,XMTDAT      ;GET TRANSMITTED CHARACTER
006662  043737  001572  001570           BIC     CARMSK,XMTDAT    ;CLEAR NON-TRANSMITTED BITS
006670  111237  001566                   MOVB    (2),RCVDAT       ;GET RECEIVED CHARACTER
006674  104002                           DATCHK                   ;COMPARE TRANS. & RCVD  CHARS

                                 ;ROUTINE TO COMPUTE AND CHECK PARITY ON RECEIVED DATA
006676  012703  000010           11$:    MOV     #8.,%3           ;GET BIT COUNTER
006702  005000                           CLR     %0               ;CLEAR COMPUTED PARITY INDICATOR
006704  106037  001566           12$:    RORB    RCVDAT           ,LOOK AT RECEIVED BIT
```

```
1611  006710  103001                        BCC     13$             ;BRANCH IF A 0
1612  006712  005100                        COM     %0              ;COMPLEMENT R0 IF A 1
1613  006714  005303                13$     DEC     %3              ;DECREMENT BIT COUNTER
1614  006716  001372                        BNE     12$             ;LOOK AT NEXT BIT IF NOT DONE
1615  006720  000240                        NOP                     ;IF COMPUTED PARITY WAS ODD R0 WILL
1616                                        ;CONTAIN ALL 1'S, IF EVEN R0 = 0
1617  006722  112237  001566                MOVB    (2)+,RCVDAT     ;GET RECEIVED CHARACTER
1618  006726  012705  000001                MOV     #1,%5           ;ROTATE PARITY BUFFER 1 PLACE
1619  006732  004537  006764                JSR     5,RORPARBUF     ;RIGHT LEAVING RECEIVED PARITY BIT IN CARRY
1620  006736  103004                        BCC     14$             ;BRANCH IF RECEIVED PARITY WAS EVEN
1621  006740  005700                        TST     %0              ;TEST FOR COMPUTED ODD PARITY
1622  006742  001336                        BNE     10$             ;BRANCH IF COMPUTED & RECEIVED WAS ODD
1623  006744  104001                        ERROR                   ;ERROR! COMPUTED =EVEN,RECEIVED = ODD
1624  006746  000734                        BR      10$             ;CONTINUE TEST
1625  006750  005700                14$     TST     %0              ;TEST FOR EVEN COMPUTED PARITY
1626  006752  001732                        BEQ     10$             ;BRANCH IF COMPUTED PARITY WAS EVEN
1627  006754  104001                        ERROR                   ;ERROR! COMPUTED =ODD,RECEIVED = EVEN
1628  006756  000730                        BR      10$             ;CONTINUE TEST
1629  006760  005726                15$     POPSP                   ;REPOSITION STACK POINTER
1630  006762  104006                        SCOPE                   ;SCOPE
1631
1632                                        ;ROUTINE TO ROTATE PARITY BUFFER.
1633  006764  006037  007026        RORPARBUF: ROR  PAR0
1634  006770  006037  007030                ROR     PAR1
1635  006774  006037  007032                ROR     PAR2
1636  007000  006037  007034                ROR     PAR3
1637  007004  006037  007036                ROR     PAR4
1638  007010  006037  007040                ROR     PAR5
1639  007014  006037  007042                ROR     PAR6
1640  007020  005316                        DEC     (SP)            ;DECREMENT ROTATE COUNT
1641  007022  001360                        BNE     RORPARBUF
1642  007024  000205                        RTS     5
1643                                        ;PARITY BUFFER
1644  007026  000000                PAR0:   OPEN
1645  007030  000000                PAR1:   OPEN
1646  007032  000000                PAR2:   OPEN
1647  007034  000000                PAR3:   OPEN
1648  007036  000000                PAR4:   OPEN
1649  007040  000000                PAR5:   OPEN
1650  007042  000000                PAR6:   OPEN
1651
1652                                        ;SUBROUTINE TO FORM LINE BIT POSITION WITH THE LINE # IN LINE
1653  007044  010046                GTLINB. MOV     %0,-(SP)        ;SAVE R0 ON THE STACK
1654  007046  005037  001564                CLR     @#LINBIT        ;CLEAR LINE BIT
1655  007052  013700  016730                MOV     @#LINE,%0       ;GET LINE
1656  007056  000261                        SEC                     ;SET CARRY
1657  007060  006137  001564        1$:     ROL     LINBIT          ;SHIFT LINE BIT
1658  007064  162700  000002                SUB     #2,%0           ;SUBTRACT 2 FROM LINE NUMBER
1659  007070  100373                        BPL     1$              ;BRANCH IF GREATER THAN 0
1660  007072  012600                        MOV     (SP)+,%0        ;RESTORE R0
1661  007074  000207                        RTS     7               ;EXIT SUBROUTINE
1662
```

```
1663
1664  007076  104000                  PRGO:   TYPE
1665  007100  020103                          PRGOM
1666  007102  012737  007136  002000          MOV     #RTO,KSTART     ;GET ADDRESS OF FIRST TEST
1667  007110  005037  002004                  CLR     RTNNO           ;CLEAR ROUTINE #
1668  007114  000137  002546                  JMP     SRSET
1669  007120  012737  007136  002000  PRGOR:  MOV     #RTO,KSTART     ;GET ADDRESS OF FIRST TEST
1670  007126  005037  002004                  CLR     RTNNO           ;CLEAR ROUTINE NUMBER
1671  007132  000137  002600                  JMP     GETRDY          ;GO AND START PROGRAM
1672                                  ;*****************************************************************
1673  007136  000000                  RTO.    0                       ;ROUTINE # 0                    *
1674  007140  007200                          RT1                     ;ADDR OF NEXT ROUTINE           *
1675  007142  000144                          100.                    ;ITERATION COUNT                *
1676  007144  007146                          RTOA                    ;SCOPE ENTRY POINT              *
1677          000000                          X=X+1
1678                                  ;*****************************************************************
1679
1680                                  ;TEST ABILITY TO REFERENCE CSR WITHOUT TRAPPING
1681  007146  012737  007170  000004  RTOA:   MOV     #1$,@#ERRVEC    ;SET UP ERROR TRAP
1682  007154  005777  172362                  TST     @CSR            ;REFERENCE CSR
1683  007160  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC      ;RESET TIME OUT TRAP
1684  007166  104006                          SCOPE
1685  007170  162716  000004          1$:     SUB     #4,(6)          ;RESTORE PC TO WHERE THE ILLEGAL
1686                                                                  ;REFERENCE OCCURED
1687  007174  104001                          ERROR                   ;ERROR! ILLEGAL REFERENCE OCCURED
1688  007176  000002                          RTI                     ;LOOP ILLEGAL REFERENCE INSTRUCTION
1689                                  ;*****************************************************************
1690  007200  000001                  RT1     1                       ;ROUTINE # 1                    *
1691  007202  007252                          RT2                     ;ADDR OF NEXT ROUTINE.          *
1692  007204  000144                          100.                    ;ITERATION COUNT                *
1693  007206  007210                          RT1A                    ;SCOPE ENTRY POINT.             *
1694          000001                          X=X+1
1695                                  ;*****************************************************************
1696
1697                                  ;TEST THAT CSR BIT0 CAN BE SET AND CLEARED
1698  007210  012777  000001  172324  RT1A:   MOV     #BIT0,@CSR      ;SET BIT0.
1699  007216  022777  000001  172316          CMP     #BIT0,@CSR      ;TEST THAT BIT0 IS SET
1700  007224  001402                          BEQ     1$              ;BRANCH IF SET
1701  007226  104001                          ERROR                   ;CSR BIT0 FAILED TO SET
1702  007230  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1703  007232  042777  000001  172302  1$:     BIC     #BIT0,@CSR      ;CLEAR BIT0
1704  007240  005777  172276                  TST     @CSR            ;TEST THAT BIT0 IS CLEAR
1705  007244  001401                          BEQ     2$
1706  007246  104001                          ERROR                   ;CSR BIT0 FAILED TO CLEAR
1707  007250  104006                  2$:     SCOPE
1708                                  ;*****************************************************************
1709  007252  000002                  RT2:    2                       ;ROUTINE # 2                    *
1710  007254  007324                          RT3                     ;ADDR OF NEXT ROUTINE.          *
1711  007256  000144                          100.                    ;ITERATION COUNT                *
1712  007260  007262                          RT2A                    ;SCOPE ENTRY POINT.             *
1713          000002                          X=X+1
1714                                  ;*****************************************************************
1715
1716                                  ;TEST THAT CSR BIT1 CAN BE SET AND CLEARED
1717  007262  012777  000002  172252  RT2A:   MOV     #BIT1,@CSR      ;SET BIT1.
1718  007270  022777  000002  172244          CMP     #BIT1,@CSR      ;TEST THAT BIT1 IS SET
```

```
1719  007276  001402                        BEQ     1$              ;BRANCH IF SET
1720  007300  104001                        ERROR                   ;CSR BIT1 FAILED TO SET
1721  007302  000407                        BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1722  007304  042777  000002  172230  1$:   BIC     #BIT1,@CSR      ;CLEAR BIT1
1723  007312  005777  172224                TST     @CSR            ;TEST THAT BIT1 IS CLEAR
1724  007316  001401                        BEQ     2$
1725  007320  104001                        ERROR                   ;CSR BIT1 FAILED TO CLEAR
1726  007322  104006                2$:     SCOPE
1727                                ;*****************************************************************************
1728  007324  000003                RT3:    3                       ;ROUTINE # 3                                *
1729  007326  007376                        RT4                     ;ADDR OF NEXT ROUTINE                       *
1730  007330  000144                        100.                    ;ITERATION COUNT                            *
1731  007332  007334                        RT3A                    ;SCOPE ENTRY POINT.                         *
1732          000003                        X=X+1
1733                                ;*****************************************************************************
1734
1735                                ;TEST THAT CSR BIT2 CAN BE SET AND CLEARED
1736  007334  012777  000004  172200  RT3A:  MOV     #BIT2,@CSR      ;SET BIT2.
1737  007342  022777  000004  172172        CMP     #BIT2,@CSR      ;TEST THAT BIT2 IS SET
1738  007350  001402                        BEQ     1$              ;BRANCH IF SET
1739  007352  104001                        ERROR                   ;CSR BIT2 FAILED TO SET
1740  007354  000407                        BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1741  007356  042777  000004  172156  1$:   BIC     #BIT2,@CSR      ;CLEAR BIT2
1742  007364  005777  172152                TST     @CSR            ;TEST THAT BIT2 IS CLEAR
1743  007370  001401                        BEQ     2$
1744  007372  104001                        ERROR                   ;CSR BIT2 FAILED TO CLEAR
1745  007374  104006                2$:     SCOPE
1746                                ;*****************************************************************************
1747  007376  000004                RT4:    4                       ;ROUTINE # 4                                *
1748  007400  007450                        RT5                     ;ADDR OF NEXT ROUTINE                       *
1749  007402  000144                        100.                    ;ITERATION COUNT                            *
1750  007404  007406                        RT4A                    ;SCOPE ENTRY POINT.                         *
1751          000004                        X=X+1
1752                                ;*****************************************************************************
1753
1754                                ;TEST THAT CSR BIT4 CAN BE SET AND CLEARED
1755  007406  012777  000020  172126  RT4A:  MOV     #BIT4,@CSR      ;SET BIT4.
1756  007414  022777  000020  172120        CMP     #BIT4,@CSR      ;TEST THAT BIT4 IS SET
1757  007422  001402                        BEQ     1$              ;BRANCH IF SET
1758  007424  104001                        ERROR                   ;CSR BIT4 FAILED TO SET
1759  007426  000407                        BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1760  007430  042777  000020  172104  1$.   BIC     #BIT4,@CSR      ;CLEAR BIT4
1761  007436  005777  172100                TST     @CSR            ;TEST THAT BIT4 IS CLEAR
1762  007442  001401                        BEQ     2$
1763  007444  104001                        ERROR                   ;CSR BIT4 FAILED TO CLEAR
1764  007446  104006                2$:     SCOPE
1765                                ;*****************************************************************************
1766  007450  000005                RT5:    5                       ;ROUTINE # 5                                *
1767  007452  007522                        RT6                     ;ADDR OF NEXT ROUTINE.                      *
1768  007454  000144                        100.                    ;ITERATION COUNT                            *
1769  007456  007460                        RT5A                    ;SCOPE ENTRY POINT.                         *
1770          000005                        X=X+1
1771                                ;*****************************************************************************
1772
1773                                ;TEST THAT CSR BIT5 CAN BE SET AND CLEARED
1774  007460  012777  000040  172054  RT5A:  MOV     #BIT5,@CSR      ;SET BIT5.
```

```
1775  007466  022777  000040  172046          CMP     #BIT5,@CSR      ;TEST THAT BIT5 IS SET
1776  007474  001402                          BEQ     1$              ;BRANCH IF SET
1777  007476  104001                          ERROR                   ;CSR BIT5 FAILED TO SET
1778  007500  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1779  007502  042777  000040  172032  1$:     BIC     #BIT5,@CSR      ;CLEAR BIT5
1780  007510  005777  172026                  TST     @CSR            ;TEST THAT BIT5 IS CLEAR
1781  007514  001401                          BEQ     2$
1782  007516  104001                          ERROR                   ;CSR BIT5 FAILED TO CLEAR
1783  007520  104006                  2$:     SCOPE
1784                                  ;*****************************************************************************
1785  007522  000006                  RT6:    6                       ;ROUTINE # 6                                *
1786  007524  007574                          RT7                     ;ADDR OF NEXT ROUTINE                       *
1787  007526  000144                          100.                    ;ITERATION COUNT                            *
1788  007530  007532                          RT6A                    ;SCOPE ENTRY POINT.                         *
1789          000006                          X=X+1
1790                                  ;*****************************************************************************
1791
1792                                  ;TEST THAT CSR BIT6 CAN BE SET AND CLEARED
1793  007532  012777  000100  172002  RT6A:   MOV     #BIT6,@CSR      ;SET BIT6.
1794  007540  022777  000100  171774          CMP     #BIT6,@CSR      ;TEST THAT BIT6 IS SET
1795  007546  001402                          BEQ     1$              ;BRANCH IF SET
1796  007550  104001                          ERROR                   ;CSR BIT6 FAILED TO SET
1797  007552  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1798  007554  042777  000100  171760  1$:     BIC     #BIT6,@CSR      ;CLEAR BIT6
1799  007562  005777  171754                  TST     @CSR            ;TEST THAT BIT6 IS CLEAR
1800  007566  001401                          BEQ     2$
1801  007570  104001                          ERROR                   ;CSR BIT6 FAILED TO CLEAR
1802  007572  104006                  2$:     SCOPE
1803                                  ;*****************************************************************************
1804  007574  000007                  RT7:    7                       ;ROUTINE # 7                                *
1805  007576  007646                          RT10                    ;ADDR OF NEXT ROUTINE                       *
1806  007600  000144                          100.                    ;ITERATION COUNT                            *
1807  007602  007604                          RT7A                    ;SCOPE ENTRY POINT.                         *
1808          000007                          X=X+1
1809                                  ;*****************************************************************************
1810
1811                                  ;TEST THAT CSR BIT12 CAN BE SET AND CLEARED
1812  007604  012777  010000  171730  RT7A:   MOV     #BIT12,@CSR     ;SET BIT12.
1813  007612  022777  010000  171722          CMP     #BIT12,@CSR     ;TEST THAT BIT12 IS SET
1814  007620  001402                          BEQ     1$              ;BRANCH IF SET
1815  007622  104001                          ERROR                   ;CSR BIT12 FAILED TO SET
1816  007624  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
1817  007626  042777  010000  171706  1$:     BIC     #BIT12,@CSR     ;CLEAR BIT12
1818  007634  005777  171702                  TST     @CSR            ;TEST THAT BIT12 IS CLEAR
1819  007640  001401                          BEQ     2$
1820  007642  104001                          ERROR                   ;CSR BIT12 FAILED TO CLEAR
1821  007644  104006                  2$:     SCOPE
1822                                  ;*****************************************************************************
1823  007646  000010                  RT10:   10                      ;ROUTINE # 10                               *
1824  007650  007720                          RT11                    ;ADDR OF NEXT ROUTINE.                      *
1825  007652  000144                          100.                    ;ITERATION COUNT                            *
1826  007654  007656                          RT10A                   ;SCOPE ENTRY POINT.                         *
1827          000010                          X=X+1
1828                                  ;*****************************************************************************
1829
1830                                  ;TEST THAT CSR BIT13 CAN BE SET AND CLEARED
```

```
 1831  007656  012777  020000  171656  RT10A:  MOV     #BIT13,@CSR     ;SET BIT13.
 1832  007664  022777  020000  171650          CMP     #BIT13,@CSR     ;TEST THAT BIT13 IS SET
 1833  007672  001402                          BEQ     1$              ;BRANCH IF SET
 1834  007674  104001                          ERROR                   ;CSR BIT13 FAILED TO SET
 1835  007676  000407                          BR      2$              ;OR AN ADDITIONAL BIT ALSO SET
 1836  007700  042777  020000  171634  1$:     BIC     #BIT13,@CSR     ;CLEAR BIT13
 1837  007706  005777  171630                  TST     @CSR            ;TEST THAT BIT13 IS CLEAR
 1838  007712  001401                          BEQ     2$
 1839  007714  104001                          ERROR                   ;CSR BIT13 FAILED TO CLEAR
 1840  007716  104006                  2$.     SCOPE
 1841                                  ;*********************************************************************
 1842  007720  000011                  RT11:   11                      ;ROUTINE # 11                      *
 1843  007722  010010                          RT12                    ;ADDR OF NEXT ROUTINE              *
 1844  007724  000144                          100.                    ;ITERATION COUNT                   *
 1845  007726  007730                          RT11A                   ;SCOPE ENTRY POINT                 *
 1846          000011                          X=X+1
 1847                                  ;*********************************************************************
 1848
 1849                                  ;TEST THAT RESET & CLEAR INSTRUCTION CLEAR ALL R/W BITS IN THE CONTROL
 1850                                  ; STATUS REG. (CSR)
 1851  007730  012777  030167  171604  RT11A:  MOV     #30167,@CSR     ;SET ALL R/W BITS IN THE CSR
 1852  007736  005037  001570                  CLR     XMTDAT
 1853  007742  104005                          SRESET                  ;ISSUE RESET
 1854  007744  017737  171572  001566          MOV     @CSR,RCVDAT     ;GET CSR CONTENTS
 1855  007752  001402                          BEQ     1$              ;BRANCH IF RESET CLEARED ALL BITS
 1856  007754  104011                          ERROR1                  ;ERROR! RESET DID NOT CLEAR ALL BITS
 1857  007756  000764                          BR      RT11A           ;LOOP ON ERROR
 1858  007760  012777  030167  171554  1$:     MOV     #30167,@CSR     ;SET ALL R/W BITS IN CSR
 1859  007766  005077  171550                  CLR     @CSR            ;CLEAR THE CSR
 1860  007772  017737  171544  001566          MOV     @CSR,RCVDAT     ;GET & TEST CSR
 1861  010000  001402                          BEQ     2$              ;GO TO EXIT IF RESULT = 0
 1862  010002  104011                          ERROR1                  ;ERROR! CLEAR INST. DID NOT CLEAR ALL BITS
 1863  010004  000765                          BR      1$              ;LOOP ERROR
 1864  010006  104006                  2$:     SCOPE                   ;SCOPE
 1865                                  ;*********************************************************************
 1866  010010  000012                  RT12:   12                      ;ROUTINE # 12                      *
 1867  010012  010144                          RT13                    ;ADDR OF NEXT ROUTINE.             *
 1868  010014  000012                          10.                     ;ITERATION COUNT                   *
 1869  010016  010020                          RT12A                   ;SCOPE ENTRY POINT.                *
 1870          000012                          X=X+1
 1871                                  ;*********************************************************************
 1872
 1873                                  ;TEST THAT A BINARY COUNT CAN BE LOADED INTO A CLEAR BKCSR AND THAT
 1874                                  ;A BINARY COUNT CAN BE CLEARED.
 1875  010020  005037  001570          RT12A:  CLR     XMTDAT
 1876  010024  013777  001570  171514  1$:     MOV     XMTDAT,@BKCSR   ;LOAD BINARY COUNT INTO BKCSR
 1877  010032  017737  171510  001566          MOV     @BKCSR,RCVDAT   ;GET BKCSR DATA
 1878  010040  023737  001570  001566          CMP     XMTDAT,RCVDAT   ;COMPARE DATA LOADED & DATA READ BACK
 1879  010046  001405                          BEQ     2$              ;BRANCH IF DATA COMPARES
 1880  010050  104011                          ERROR1                  ;ERROR! DATA DOD NIT COMPARE
 1881  010052  032777  040000  171122          BIT     #BIT14,@SWR     ;SCOPE LOOP?
 1882  010060  001361                          BNE     1$              ;BRANCH IF SCOPE LOOP
 1883  010062  013701  001570          2$:     MOV     XMTDAT,X1       ;SAVE BINARY COUNT
 1884  010066  005037  001570                  CLR     XMTDAT
 1885  010072  005077  171450          3$:     CLR     @BKCSR          ;CLEAR BKCSR AND TEST
 1886  010076  017737  171444  001566          MOV     @BKCSR,RCVDAT   ;BKCSR CAN BE CLEARED
```

```
1887  010104  001405                          BEQ     4$              ;BRANCH IF BKCSR CLEARED
1888  010106  104011                          ERROR1                  ;ERROR! BKCSR DID NOT CLEAR
1889  010110  032777  040000  171064          BIT     @#BIT14,@#SWR   ;SCOPE LOOP?
1890  010116  001365                          BNE     3$              ;BRANCH IF SCOPE LOOP
1891  010120  010137  001570          4$:     MOV     %1,XMTDAT       ;GET BINARY COUNT
1892  010124  023727  001570  177777          CMP     XMTDAT,#-1      ;ALL NUMBERS BEEN LOADED
1893  010132  001403                          BEQ     5$              ;GO TO EXIT
1894  010134  005237  001570                  INC     XMTDAT          ;INCREMENT BINARY COUNT
1895  010140  000731                          BR      1$              ;REPEAT TEST
1896  010142  104006                  5$:     SCOPE                   ;SCOPE
1897                                  ;*****************************************************************************
1898  010144  000013                  RT13:   13                      ;ROUTINE # 13                              *
1899  010146  010256                          RT14                    ;ADDR OF NEXT ROUTINE.                     *
1900  010150  000144                          100.                    ;ITERATION COUNT                           *
1901  010152  010154                          RT13A                   ;SCOPE ENTRY POINT.                        *
1902          000013                          X=X+1
1903                                  ;*****************************************************************************
1904
1905                                  ;TEST THAT RANDOM NUMBERS CAN BE LOADED INTO THE BKCSR
1906  010154  012702  010000          RT13A:  MOV     #10000,%2       ;GET RANDOM #COUNTER
1907  010160  017737  171362  002104  1$:     MOV     @#BKCSR,PRVCNT  ;GET PREVIOUS CONTENTS
1908  010166  004737  003250                  JSR     7,@#RMGEN       ;GO GET A RANDOM NUMBER
1909  010172  010037  001570                  MOV     %0,XMTDAT       ;GET RANDOM NUMBER
1910  010176  013777  001570  171342  2$:     MOV     XMTDAT,@#BKCSR  ;LOAD RANDOM NUMBER INTO BKCSR
1911  010204  017737  171336  001566          MOV     @#BKCSR,RCVDAT  ;GET BKCSR DATA
1912  010212  023737  001570  001566          CMP     XMTDAT,RCVDAT   ;COMPARE DATA
1913  010220  001401                          BEQ     3$              ;BRANCH IF SAME
1914  010222  104011                          ERROR1                  ;ERROR! DATA NOT THE SAME
1915  010224  032777  040000  170750  3$:     BIT     @#BIT14,@#SWR   ;SCOPE LOOP?
1916  010232  001406                          BEQ     4$              ;BRANCH IF NO LOOP ON ERROR
1917  010234  005077  171306                  CLR     @#BKCSR         ;
1918  010240  013777  002104  171300          MOV     PRVCNT,@#BKCSR  ;LOAD PREVIOUS CONTENTS
1919  010246  000753                          BR      2$              ;REPEAT TEST
1920  010250  005302                  4$:     DEC     %2
1921  010252  001342                          BNE     1$              ;BRANCH IF NOT
1922  010254  104006                  5$:     SCOPE                   ;SCOPE
1923                                  ;*****************************************************************************
1924  010256  000014                  RT14:   14                      ;ROUTINE # 14                              *
1925  010260  010316                          RT15                    ;ADDR OF NEXT ROUTINE.                     *
1926  010262  000012                          10.                     ;ITERATION COUNT                           *
1927  010264  010266                          RT14A                   ;SCOPE ENTRY POINT.                        *
1928          000014                          X=X+1
1929                                  ;*****************************************************************************
1930
1931                                  ;TEST THAT RESET CLEARS ALL BREAK STATUS REGISTER BITS
1932
1933  010266  012777  177777  171252  RT14A:  MOV     #-1,@#BKCSR
1934  010274  005037  001570                  CLR     XMTDAT
1935  010300  104005                          SRESET
1936  010302  017737  171240  001566          MOV     @#BKCSR,RCVDAT
1937  010310  001401                          BEQ     1$
1938  010312  104011                          ERROR1
1939  010314  104006                  1$:     SCOPE
1940                                  ;*****************************************************************************
1941  010316  000015                  RT15:   15                      ;ROUTINE # 15                              *
1942  010320  010452                          RT16                    ;ADDR OF NEXT ROUTINE                      *
```

```
1943  010322  000012                          10.                     ;ITERATION COUNT                        *
1944  010324  010326                          RT15A                   ;SCOPE ENTRY POINT.                     *
1945          000015                          X=X+1
1946                                  ;*****************************************************************************
1947
1948                                  ;TEST THAT A BINARY COUNT CAN BE LOADED INTO A CLEAR BASREG AND THAT
1949                                  ;A BINARY COUNT CAN BE CLEARED.
1950  010326  005037  001570          RT15A:  CLR     XMTDAT
1951  010332  013777  001570  171210  1$:     MOV     XMTDAT,@BASREG  ;LOAD BINARY COUNT INTO BASREG
1952  010340  017737  171204  001566          MOV     @BASREG,RCVDAT  ;GET BASREG DATA
1953  010346  023737  001570  001566          CMP     XMTDAT,RCVDAT   ;COMPARE DATA
1954  010354  001405                          BEQ     2$              ;BRANCH IF DATA COMPARES
1955  010356  104011                          ERROR1                  ;ERROR! DATA DID NOT COMPARE
1956  010360  032777  040000  170614          BIT     #BIT14,@SWR     ;SCOPE LOOP?
1957  010366  001361                          BNE     1$              ;BRANCH IF SCOPE LOOP
1958  010370  013701  001570          2$      MOV     XMTDAT,R1       ;SAVE BINARY COUNT
1959  010374  005037  001570                  CLR     XMTDAT
1960  010400  005077  171144          3$:     CLR     @BASREG
1961  010404  017737  171140  001566          MOV     @BASREG,RCVDAT
1962  010412  001405                          BEQ     4$              ;BRANCH IF BKCSR CLEARED
1963  010414  104011                          ERROR1                  ;ERROR! BKCSR DID NOT CLEAR
1964  010416  032777  040000  170556          BIT     #BIT14,@SWR     ;SCOPE LOOP?
1965  010424  001365                          BNE     3$              ;BRANCH IF SCOPE LOOP
1966  010426  010137  001570          4$.     MOV     R1,XMTDAT       ;GET BINARY COUNT
1967  010432  023727  001570  177000          CMP     XMTDAT,#177000  ;ALL NUMBERS BEEN LOADED
1968  010440  001403                          BEQ     5$              ;GO TO EXIT
1969  010442  105237  001571                  INCB    XMTDAT+1        ;INCREMENT BINARY COUNT
1970  010446  000731                          BR      1$              ;REPEAT TEST
1971  010450  104006          5$:     SCOPE                           ;SCOPE
1972                                  ;*****************************************************************************
1973  010452  000016          RT16:   16                              ;ROUTINE # 16                           *
1974  010454  010570                  RT17                            ;ADDR OF NEXT ROUTINE.                  *
1975  010456  000144                  100.                            ;ITERATION COUNT                        *
1976  010460  010462                  RT16A                           ;SCOPE ENTRY POINT                      *
1977          000016                  X=X+1
1978                                  ;*****************************************************************************
1979
1980                                  ;TEST THAT RANDOM NUMBERS CAN BE LOADED INTO THE BASE REGISTER
1981  010462  012702  010000          RT16A:  MOV     #10000,R2       ;GET RANDOM #COUNTER
1982  010466  017737  171056  002104  1$:     MOV     @BASREG,PRVCNT  ;GET PREVIOUS CONTENTS
1983  010474  004737  003250                  JSR     7,@RANGEN       ;GO GET A RANDOM NUMBER
1984  010500  042700  000377                  BIC     #000377,R0      ;CLEAR UNUSED BITS
1985  010504  010037  001570                  MOV     R0,XMTDAT       ;GET RANDOM NUMBER
1986  010510  013777  001570  171032  2$:     MOV     XMTDAT,@BASREG  ;LOAD RANDOM NUMBER INTO BASREG
1987  010516  017737  171026  001566          MOV     @BASREG,RCVDAT  ;GET BASREG DATA
1988  010524  023737  001570  001566          CMP     XMTDAT,RCVDAT   ;COMPARE DATA
1989  010532  001401                          BEQ     3$              ;BRANCH IF SAME
1990  010534  104011                          ERROR1                  ;ERROR! DATA NOT THE SAME
1991  010536  032777  040000  170436  3$:     BIT     #BIT14,@SWR     ;SCOPE LOOP?
1992  010544  001406                          BEQ     4$              ;BRANCH IF NO LOOP ON ERROR
1993  010546  005077  170776                  CLR     @BASREG
1994  010552  013777  002104  170770          MOV     PRVCNT,@BASREG  ;LOAD PREVIOUS CONTENTS
1995  010560  000753                          BR      2$              ;REPEAT TEST
1996  010562  005302          4$:     DEC     R2              ;10000 NUMBERS BEEN TESTED
1997  010564  001340                  BNE     1$              ;BRANCH IF NOT
1998  010566  104006          5$:     SCOPE                           ;SCOPE
```

```
                                        ;*************************************************************
        010570  000017                  RT17:   17                      ;ROUTINE # 17                       *
        010572  010662                          RT20                    ;ADDR OF NEXT ROUTINE               *
        010574  000144                          100.                    ;ITERATION COUNT                    *
        010576  010600                          RT17A                   ;SCOPE ENTRY POINT.                 *
                000017                          X=X+1
                                        ;*************************************************************

                                        ;TEST THAT ALL BAR BITS CAN BE INDIVIDUALLY SET AND CLEARED
        010600  013701  001544          RT17A:  MOV     BAR,X1          ;GET BAR ADDRESS
        010604  012777  001400  170736          MOV     #CAT,@BASREG    ;INITIALIZE BASE REGISTER
        010612  012700  000001                  MOV     #1,X0           ;GET BIT TESTER
        010616  050011                  1$      BIS     X0,(1)          ;SET BAR BIT
        010620  020011                          CMP     X0,(1)          ;TEST THAT ONLY THE PROPER BAR BIT SET
        010622  001006                          BNE     3$              ;BRANCH IF ERROR
        010624  040011                          BIC     X0,(1)          ;CLEAR BAR BIT
        010626  005711                          TST     (1)             ;TEST THAT BAR BIT CLEARED
        010630  001011                          BNE     5$              ;BRANCH IF BAR BIT FAILED TO CLEAR
        010632  006300                          ASL     X0              ;SHIFT BIT TESTER
        010634  103370                          BCC     1$
        010636  104006                  2$.     SCOPE                   ;SCOPE
        010640  010037  001570          3$:     MOV     X0,XMTDAT       ;GET WHAT DATA WAS SUPPOSED TO BE
        010644  011137  001566          4$:     MOV     (1),RCVDAT      ;GET WHAT DATA WAS
        010650  104011                          ERROR1                  ;ERROR! IMPROPER BIT OPERATION
        010652  000771                          BR      2$              ;GO TO SCOPE
        010654  005037  001570          5$:     CLR     XMTDAT          ;GET WHAT DATA WAS SUPPOSED TO BE
        010660  000771                          BR      4$
                                        ;*************************************************************
        010662  000020                  RT20:   20                      ;ROUTINE # 20                       *
        010664  010750                          RT21                    ;ADDR OF NEXT ROUTINE.              *
        010666  000012                          10.                     ;ITERATION COUNT                    *
        010670  010672                          RT20A                   ;SCOPE ENTRY POINT.                 *
                000020                          X=X+1
                                        ;*************************************************************

                                        ;TEST THAT RESET CLEARS ALL BAR BITS
        010672  005037  001570          RT20A:  CLR     XMTDAT
        010676  052777  177777  170640          BIS     #-1,@BAR        ;SET ALL BAR BITS
        010704  104005                          SRESET                  ;RESET
        010706  017737  170632  001566          MOV     @BAR,RCVDAT     ;GET BAR DATA
        010714  001402                          BEQ     1$              ;BRANCH IF ALL 0'S
        010716  104011                          ERROR1                  ;ERROR! RESET DID NOT CLEAR ALL BAR BITS
        010720  000412                          BR      2$              ;GO TO EXIT
        010722  052777  177777  170614  1$:     BIS     #-1,@BAR        ;SET ALL BIT IN THE BAR
        010730  005077  170610                  CLR     @BAR            ;CLEAR ALL BITS IN THE BAR
        010734  017737  170604  001566          MOV     @BAR,RCVDAT     ;GET & TEST RESULT OF CLEAR OPERATION
        010742  001401                          BEQ     2$              ;EXIT IF ALL BITS CLEARED
        010744  104011                          ERROR1                  ;ERROR! ALL BITS DID NOT CLEAR
        010746  104006                  2$:     SCOPE                   ;SCOPE
                                        ;*************************************************************
        010750  000021                  RT21:   21                      ;ROUTINE # 21                       *
        010752  011056                          RT22                    ;ADDR OF NEXT ROUTINE.              *
        010754  000144                          100.                    ;ITERATION COUNT                    *
        010756  010760                          RT21A                   ;SCOPE ENTRY POINT.                 *
                000021                          X=X+1
                                        ;*************************************************************
```

```
 2055
 2056                                          ;TEST THAT CSR RESPONDS PROPERLY TO BYTE COMMANDS
 2057  010760  012777  010100  170554  RT21A:  MOV     #10100,@CSR     ;LOAD TEST NUMBER IN CSR
 2058  010766  105077  170550                  CLRB    @CSR            ;CLEAR EVEN BYTE
 2059  010772  022777  010000  170542          CMP     #10000,@CSR     ;TEST THAT ONLY EVEN BYTE CLEARED
 2060  011000  001410                          BEQ     1$
 2061  011002  012737  010100  001570          MOV     #10100,XMTDAT   ;LOAD CORRECT RESULT
 2062  011010  017737  170526  001566          MOV     @CSR,RCVDAT     ;GET ACTUAL RESULT
 2063  011016  104011                          ERROR1                  ;ERROR! EVEN BYTE INSTRUCTION FAILED
 2064  011020  000415                          BR      2$              ;GO TO SCOPE
 2065  011022  012777  010100  170512  1$.     MOV     #10100,@CSR     ;LOAD TEST NUMBER IN CSR
 2066  011030  105077  171052                  CLRB    @.CSR           ;TEST THAT ONLY ODD BYTE CLEARED
 2067  011034  001407                          BEQ     2$
 2068  011036  012737  000100  001570          MOV     #00100,XMTDAT   ;LOAD CORRECT RESULT
 2069  011044  017737  170472  001566          MOV     @CSR,RCVDAT     ;LOAD ACTUAL RESULT
 2070  011052  104011                          ERROR1                  ;ERROR! ODD BYTE INSTRUCTION FAILED
 2071  011054  104006                  2$:     SCOPE                   ;SCOPE
 2072                                          ;*****************************************************************************
 2073  011056  000022                  RT22:   22                      ;ROUTINE # 22                                    *
 2074  011060  011164                          RT23                    ;ADDR OF NEXT ROUTINE                            *
 2075  011062  000144                          100.                    ;ITERATION COUNT                                 *
 2076  011064  011066                          RT22A                   ;SCOPE ENTRY POINT.                              *
 2077          000022                          X=X+1
 2078                                          ;*****************************************************************************
 2079
 2080                                          ;TEST THAT BAR RESPONDS PROPERLY TO BYTE COMMANDS
 2081  011066  012777  010100  170450  RT22A:  MOV     #10100,@BAR     ;LOAD TEST NUMBER IN BAR
 2082  011074  105077  170444                  CLRB    @BAR            ;CLEAR EVEN BYTE
 2083  011100  022777  010000  170436          CMP     #10000,@BAR     ;TEST THAT ONLY EVEN BYTE CLEARED
 2084  011106  001410                          BEQ     1$
 2085  011110  012737  010100  001570          MOV     #10100,XMTDAT   ;LOAD CORRECT RESULT
 2086  011116  017737  170422  001566          MOV     @BAR,RCVDAT     ;GET ACTUAL RESULT
 2087  011124  104011                          ERROR1                  ;ERROR! EVEN BYTE INSTRUCTION FAILED
 2088  011126  000415                          BR      2$              ;GO TO SCOPE
 2089  011130  012777  010100  170406  1$:     MOV     #10100,@BAR     ;LOAD TEST NUMBER IN BAR
 2090  011136  105077  170746                  CLRB    @.BAR           ;TEST THAT ONLY ODD BYTE CLEARED
 2091  011142  001407                          BEQ     2$
 2092  011144  012737  000100  001570          MOV     #00100,XMTDAT   ;LOAD CORRECT RESULT
 2093  011152  017737  170364  001566          MOV     @CSR,RCVDAT     ;LOAD ACTUAL RESULT
 2094  011160  104011                          ERROR1                  ;ERROR! ODD BYTE INSTRUCTION FAILED
 2095  011162  104006                  2$:     SCOPE                   ;SCOPE
 2096                                          ;*****************************************************************************
 2097  011164  000023                  RT23:   23                      ;ROUTINE # 23                                    *
 2098  011166  011272                          RT24                    ;ADDR OF NEXT ROUTINE.                           *
 2099  011170  000144                          100.                    ;ITERATION COUNT                                 *
 2100  011172  011174                          RT23A                   ;SCOPE ENTRY POINT.                              *
 2101          000023                          X=X+1
 2102                                          ;*****************************************************************************
 2103
 2104                                          ;TEST THAT BKCSR RESPONDS PROPERLY TO BYTE COMMANDS
 2105  011174  012777  010100  170344  RT23A:  MOV     #10100,@BKCSR   ;LOAD TEST NUMBER IN BKCSR
 2106  011202  105077  170340                  CLRB    @BKCSR          ;CLEAR EVEN BYTE
 2107  011206  022777  010000  170332          CMP     #10000,@BKCSR   ;TEST THAT ONLY EVEN BYTE CLEARED
 2108  011214  001410                          BEQ     1$
 2109  011216  012737  010100  001570          MOV     #10100,XMTDAT   ;LOAD CORRECT RESULT
 2110  011224  017737  170316  001566          MOV     @BKCSR,RCVDAT   ;GET ACTUAL RESULT
```

```
 2111  011232  104011                          ERROR1                  ;ERROR! EVEN BYTE INSTRUCTION FAILED
 2112  011234  000415                          BR      2$              ;GO TO SCOPE
 2113  011236  012777  010100  170302  1$:     MOV     #10100,@BKCSR   ;LOAD TEST NUMBER IN BKCSR
 2114  011244  105077  170642                  CLRB    @.BKCSR         ;TEST THAT ONLY ODD BYTE CLEARED
 2115  011250  001407                          BEQ     2$
 2116  011252  012737  000100  001570          MOV     #00100,XMTDAT   ;LOAD CORRECT RESULT
 2117  011260  017737  170256  001566          MOV     @CSR,RCVDAT     ;LOAD ACTUAL RESULT
 2118  011266  104011                          ERROR1                  ;ERROR! ODD BYTE INSTRUCTION FAILED
 2119  011270  104006                  2$:     SCOPE                   ;SCOPE
 2120                                  ;*****************************************************************************
 2121  011272  000024                  RT24:   24                      ;ROUTINE # 24                              *
 2122  011274  011336                          RT25                    ;ADDR OF NEXT ROUTINE.                     *
 2123  011276  000144                          100.                    ;ITERATION COUNT                           *
 2124  011300  011302                          RT24A                   ;SCOPE ENTRY POINT.                        *
 2125          000024                          X=X+1
 2126                                  ;*****************************************************************************
 2127
 2128                                  ;TEST THAT OVER RUN BIT (CSR BIT13) CAUSES AN INTERRUPT WHEN SET
 2129  011302  012777  011334  170246  RT24A:  MOV     #1$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
 2130  011310  012777  010000  170224          MOV     #BIT12,@CSR     ;SET TRANSMITTER IE BIT
 2131  011316  052777  020000  170216          BIS     #BIT13,@CSR     ;SET OVER RUN BIT
 2132  011324  005037  177776                  CLR     @#PSW           ;ENABLE INTERRUPTS
 2133  011330  000240                          NOP
 2134  011332  104001                          ERROR                   ;ERROR!OVERRUN FAILED TO CAUSE AN
 2135                                          ;INTERRUPT,OR INTERRUPTED TO INCOR-
 2136                                          ;RECT ADDRESS
 2137  011334  104006                  1$:     SCOPE                   ;SCOPE
 2138                                  ;*****************************************************************************
 2139  011336  000025                  RT25:   25                      ;ROUTINE # 25                              *
 2140  011340  011444                          RT26                    ;ADDR OF NEXT ROUTINE                      *
 2141  011342  000100                          100                     ;ITERATION COUNT                           *
 2142  011344  011346                          RT25A                   ;SCOPE ENTRY POINT                         *
 2143          000025                          X=X+1
 2144                                  ;*****************************************************************************
 2145
 2146                                  ;TEST THAT THE DM11 INTERRUPTS AT THE CORRECT LEVEL
 2147  011346  012737  000340  177776  RT25A:  MOV     #PRTY7,@#PSW
 2148  011354  012777  011404  170174          MOV     #1$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
 2149  011362  012777  030000  170152          MOV     #30000,@CSR     ;SET OVER RUN & IE BITS
 2150  011370  012737  000200  177776          MOV     #PRTY4,@#PSW    ;ALLOW INTERRUPTS ON LEVEL 5 & ABOVE
 2151  011376  000240                          NOP
 2152  011400  104001                          ERROR                   ;ERROR!DM11 FAILED TO INTERRUPT
 2153  011402  000417                          BR      3$              ;GO TO EXIT
 2154  011404  022626                  1$:     CMP     (6)+,(6)+       ;RESET STACK POINTER
 2155  011406  013737  001560  177776          MOV     XMTLVL,@#PSW    ;LOAD DM11 INTERRUPT LEVEL
 2156  011414  012777  011440  170134          MOV     #2$,@XMTINT     ;LOAD TRANSMITTER INTERRUPT VECTOR
 2157  011422  005077  170114                  CLR     @CSR
 2158  011426  012777  030000  170106          MOV     #30000,@CSR
 2159  011434  000240                          NOP
 2160  011436  000401                          BR      3$              ;GO TO EXIT
 2161  011440  104001                  2$:     ERROR                   ;ERROR! DM11 INTERRUPTED ON HIGHER
 2162                                          ;PRIORITY LEVEL THAN SET FOR
 2163  011442  104006                  3$:     SCOPE
 2164                                  ;*****************************************************************************
 2165  011444  000026                  RT26:   26                      ;ROUTINE # 26                              *
 2166  011446  011462                          RT27                    ;ADDRESS OF NEXT TEST.                     *
```

```
2167  011450  000144                    100.                    ;ITERATION COUNT                *
2168  011452  011454                    LTST0                   ;SCOPE ENTRY POINT              *
2169          000026                    X=X+1
2170                            ;*****************************************************************************
2171                            ;TRANSMITTER LINE TEST LINE 0
2172  011454  004537  004622    LTST0: JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 0
2173  011460  000000                    LINE0
2174          000001                    Y=Y+1
2175                            ;*****************************************************************************
2176  011462  000027            RT27:   27                      ;ROUTINE # 27                   *
2177  011464  011500                    RT30                    ;ADDRESS OF NEXT TEST.          *
2178  011466  000144                    100.                    ;ITERATION COUNT                *
2179  011470  011472                    LTST1                   ;SCOPE ENTRY POINT              *
2180          000027                    X=X+1
2181                            ;*****************************************************************************
2182                            ;TRANSMITTER LINE TEST LINE 1
2183  011472  004537  004622    LTST1: JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 1
2184  011476  000002                    LINE1
2185          000002                    Y=Y+1
2186                            ;*****************************************************************************
2187  011500  000030            RT30:   30                      ;ROUTINE # 30                   *
2188  011502  011516                    RT31                    ;ADDRESS OF NEXT TEST.          *
2189  011504  000144                    100.                    ;ITERATION COUNT                *
2190  011506  011510                    LTST2                   ;SCOPE ENTRY POINT              *
2191          000030                    X=X+1
2192                            ;*****************************************************************************
2193                            ;TRANSMITTER LINE TEST LINE 2
2194  011510  004537  004622    LTST2: JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 2
2195  011514  000004                    LINE2
2196          000003                    Y=Y+1
2197                            ;*****************************************************************************
2198  011516  000031            RT31:   31                      ;ROUTINE # 31                   *
2199  011520  011534                    RT32                    ;ADDRESS OF NEXT TEST.          *
2200  011522  000144                    100.                    ;ITERATION COUNT                *
2201  011524  011526                    LTST3                   ;SCOPE ENTRY POINT              *
2202          000031                    X=X+1
2203                            ;*****************************************************************************
2204                            ;TRANSMITTER LINE TEST LINE 3
2205  011526  004537  004622    LTST3: JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 3
2206  011532  000006                    LINE3
2207          000004                    Y=Y+1
2208                            ;*****************************************************************************
2209  011534  000032            RT32:   32                      ;ROUTINE # 32                   *
2210  011536  011552                    RT33                    ;ADDRESS OF NEXT TEST           *
2211  011540  000144                    100.                    ;ITERATION COUNT                *
2212  011542  011544                    LTST4                   ;SCOPE ENTRY POINT              *
2213          000032                    X=X+1
2214                            ;*****************************************************************************
2215                            ;TRANSMITTER LINE TEST LINE 4
2216  011544  004537  004622    LTST4: JSR      5,XMTTST        ;GO TEST TRANSMITTER LINE 4
2217  011550  000010                    LINE4
2218          000005                    Y=Y+1
2219                            ;*****************************************************************************
2220  011552  000033            RT33:   33                      ;ROUTINE # 33                   *
2221  011554  011570                    RT34                    ;ADDRESS OF NEXT TEST.          *
2222  011556  000144                    100.                    ;ITERATION COUNT                *
```

```
2223  011560  011562                        LTST5                   ;SCOPE ENTRY POINT                      *
2224          000033                        X=X+1
2225                                ;*****************************************************************************
2226                                ;TRANSMITTER LINE TEST LINE 5
2227  011562  004537  004622        LTST5: JSR       5,XMTTST       ;GO TEST TRANSMITTER LINE 5
2228  011566  000012                        LINE5
2229          000006                        Y=Y+1
2230                                ;*****************************************************************************
2231  011570  000034                RT34:   34                      ;ROUTINE # 34                           *
2232  011572  011606                        RT35                    ;ADDRESS OF NEXT TEST                   *
2233  011574  000144                        100.                    ;ITERATION COUNT                        *
2234  011576  011600                        LTST6                   ;SCOPE ENTRY POINT                      *
2235          000034                        X=X+1
2236                                ;*****************************************************************************
2237                                ;TRANSMITTER LINE TEST LINE 6
2238  011600  004537  004622        LTST6: JSR       5,XMTTST       ;GO TEST TRANSMITTER LINE 6
2239  011604  000019                        LINE6
2240          000007                        Y=Y+1
2241                                ;*****************************************************************************
2242  011606  000035                RT35:   35                      ;ROUTINE # 35                           *
2243  011610  011624                        RT36                    ;ADDRESS OF NEXT TEST                   *
2244  011612  000144                        100.                    ;ITERATION COUNT                        *
2245  011614  011616                        LTST7                   ;SCOPE ENTRY POINT                      *
2246          000035                        X=X+1
2247                                ;*****************************************************************************
2248                                ;TRANSMITTER LINE TEST LINE 7
2249  011616  004537  004622        LTST7: JSR       5,XMTTST       ;GO TEST TRANSMITTER LINE 7
2250  011622  000016                        LINE7
2251          000010                        Y=Y+1
2252                                ;*****************************************************************************
2253  011624  000036                RT36.   36                      ;ROUTINE # 36                           *
2254  011626  011642                        RT37                    ;ADDRESS OF NEXT TEST                   *
2255  011630  000144                        100.                    ;ITERATION COUNT                        *
2256  011632  011634                        LTST10                  ;SCOPE ENTRY POINT                      *
2257          000036                        X=X+1
2258                                ;*****************************************************************************
2259                                ;TRANSMITTER LINE TEST LINE 10
2260  011634  004537  004622        LTST10: JSR      5,XMTTST       ;GO TEST TRANSMITTER LINE 10
2261  011640  000020                        LINE10
2262          000011                        Y=Y+1
2263                                ;*****************************************************************************
2264  011642  000037                RT37:   37                      ;ROUTINE # 37                           *
2265  011644  011660                        RT40                    ;ADDRESS OF NEXT TEST.                  *
2266  011646  000144                        100.                    ;ITERATION COUNT                        *
2267  011650  011652                        LTST11                  ;SCOPE ENTRY POINT                      *
2268          000037                        X=X+1
2269                                ;*****************************************************************************
2270                                ;TRANSMITTER LINE TEST LINE 11
2271  011652  004537  004622        LTST11: JSR      5,XMTTST       ;GO TEST TRANSMITTER LINE 11
2272  011656  000022                        LINE11
2273          000012                        Y=Y+1
2274                                ;*****************************************************************************
2275  011660  000040                RT40:   40                      ;ROUTINE # 40                           *
2276  011662  011676                        RT41                    ;ADDRESS OF NEXT TEST                   *
2277  011664  000144                        100.                    ;ITERATION COUNT                        *
2278  011666  011670                        LTST12                  ;SCOPE ENTRY POINT                      *
```

```
2279            000040                          X=X+1
2280                                    ;*****************************************************************************
2281                                    ;TRANSMITTER LINE TEST LINE 12
2282  011670    004537  004622        LTST12: JSR     5,XMTTST        ;GO TEST TRANSMITTER LINE 12
2283  011674    000024                        LINE12
2284            000013                        Y=Y+1
2285                                    ;*****************************************************************************
2286  011676    000041                RT41:   41                      ;ROUTINE # 41                          *
2287  011700    011714                        RT42                    ;ADDRESS OF NEXT TEST.                 *
2288  011702    000144                        100.                    ;ITERATION COUNT                       *
2289  011704    011706                        LTST13                  ;SCOPE ENTRY POINT                     *
2290            000041                        X=X+1
2291                                    ;*****************************************************************************
2292                                    ;TRANSMITTER LINE TEST LINE 13
2293  011706    004537  004622        LTST13: JSR     5,XMTTST        ;GO TEST TRANSMITTER LINE 13
2294  011712    000026                        LINE13
2295            000014                        Y=Y+1
2296                                    ;*****************************************************************************
2297  011714    000042                RT42:   42                      ;ROUTINE # 42                          *
2298  011716    011732                        RT43                    ;ADDRESS OF NEXT TEST.                 *
2299  011720    000144                        100.                    ;ITERATION COUNT                       *
2300  011722    011724                        LTST14                  ;SCOPE ENTRY POINT                     *
2301            000042                        X=X+1
2302                                    ;*****************************************************************************
2303                                    ;TRANSMITTER LINE TEST LINE 14
2304  011724    004537  004622        LTST14: JSR     5,XMTTST        ;GO TEST TRANSMITTER LINE 14
2305  011730    000030                        LINE14
2306            000015                        Y=Y+1
2307                                    ;*****************************************************************************
2308  011732    000043                RT43:   43                      ;ROUTINE # 43                          *
2309  011734    011750                        RT44                    ;ADDRESS OF NEXT TEST                  *
2310  011736    000144                        100.                    ;ITERATION COUNT                       *
2311  011740    011742                        LTST15                  ;SCOPE ENTRY POINT                     *
2312            000043                        X=X+1
2313                                    ;*****************************************************************************
2314                                    ;TRANSMITTER LINE TEST LINE 15
2315  011742    004537  004622        LTST15: JSR     5,XMTTST        ;GO TEST TRANSMITTER LINE 15
2316  011746    000032                        LINE15
2317            000016                        Y=Y+1
2318                                    ;*****************************************************************************
2319  011750    000044                RT44:   44                      ;ROUTINE # 44                          *
2320  011752    011766                        RT45                    ;ADDRESS OF NEXT TEST.                 *
2321  011754    000144                        100.                    ;ITERATION COUNT                       *
2322  011756    011760                        LTST16                  ;SCOPE ENTRY POINT                     *
2323            000044                        X=X+1
2324                                    ;*****************************************************************************
2325                                    ;TRANSMITTER LINE TEST LINE 16
2326  011760    004537  004622        LTST16: JSR     5,XMTTST        ;GO TEST TRANSMITTER LINE 16
2327  011764    000034                        LINE16
2328            000017                        Y=Y+1
2329                                    ;*****************************************************************************
2330  011766    000045                RT45:   45                      ;ROUTINE # 45                          *
2331  011770    012004                        RT46                    ;ADDRESS OF NEXT TEST.                 *
2332  011772    000144                        100.                    ;ITERATION COUNT                       *
2333  011774    011776                        LTST17                  ;SCOPE ENTRY POINT                     *
2334            000045                        X=X+1
```

```
2335                                    ;*****************************************************************************
2336                                    ;TRANSMITTER LINE TEST LINE 17
2337  011776  004537  004622           LTST17: JSR     5,XMTTST        ;GO TEST TRANSMITTER LINE 17
2338  012002  000036                           LINE17
2339          000020                           Y=Y+1
2340          000000                           Y=0
2341          000000                           A=0
2342                                    ;*****************************************************************************
2343  012004  000046                   RT46:   46                      ;ROUTINE # 46                          *
2344  012006  012022                           RT47                    ;ADDRESS OF NEXT TEST                  *
2345  012010  000144                           100.                    ;ITERATION COUNT                       *
2346  012012  012014                           RCV0                    ;SCOPE ENTRY POINT                     *
2347          000046                           X=X+1
2348                                    ;*****************************************************************************
2349                                    ;RECEIVER LINE TEST LINE 0
2350  012014  004537  005034           RCV0:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 0
2351  012020  000000                           LINE0
2352          000001                           Y=Y+1
2353                                    ;*****************************************************************************
2354  012022  000047                   RT47:   47                      ;ROUTINE # 47                          *
2355  012024  012040                           RT50                    ;ADDRESS OF NEXT TEST                  *
2356  012026  000144                           100.                    ;ITERATION COUNT                       *
2357  012030  012032                           RCV1                    ;SCOPE ENTRY POINT                     *
2358          000047                           X=X+1
2359                                    ;*****************************************************************************
2360                                    ;RECEIVER LINE TEST LINE 1
2361  012032  004537  005034           RCV1:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 1
2362  012036  000002                           LINE1
2363          000002                           Y=Y+1
2364                                    ;*****************************************************************************
2365  012040  000050                   RT50:   50                      ;ROUTINE # 50                          *
2366  012042  012056                           RT51                    ;ADDRESS OF NEXT TEST                  *
2367  012044  000144                           100.                    ;ITERATION COUNT                       *
2368  012046  012050                           RCV2                    ;SCOPE ENTRY POINT                     *
2369          000050                           X=X+1
2370                                    ;*****************************************************************************
2371                                    ;RECEIVER LINE TEST LINE 2
2372  012050  004537  005034           RCV2:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 2
2373  012054  000004                           LINE2
2374          000003                           Y=Y+1
2375                                    ;*****************************************************************************
2376  012056  000051                   RT51:   51                      ;ROUTINE # 51                          *
2377  012060  012074                           RT52                    ;ADDRESS OF NEXT TEST                  *
2378  012062  000144                           100.                    ;ITERATION COUNT                       *
2379  012064  012066                           RCV3                    ;SCOPE ENTRY POINT                     *
2380          000051                           X=X+1
2381                                    ;*****************************************************************************
2382                                    ;RECEIVER LINE TEST LINE 3
2383  012066  004537  005034           RCV3:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 3
2384  012072  000006                           LINE3
2385          000004                           Y=Y+1
2386                                    ;*****************************************************************************
2387  012074  000052                   RT52:   52                      ;ROUTINE # 52                          *
2388  012076  012112                           RT53                    ;ADDRESS OF NEXT TEST                  *
2389  012100  000144                           100.                    ;ITERATION COUNT                       *
2390  012102  012104                           RCV4                    ;SCOPE ENTRY POINT                     *
```

```
                000052                          X=X+1
                                        ;*******************************************************************
                                        ;RECEIVER LINE TEST LINE 4
        012104  004537  005034          RCV4:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 4
        012110  000010                          LINE4
                000005                          Y=Y+1
                                        ;*******************************************************************
        012112  000053                  RT53:   53                      ;ROUTINE # 53              *
        012114  012130                          RT54                    ;ADDRESS OF NEXT TEST      *
        012116  000144                          100.                    ;ITERATION COUNT           *
        012120  012122                          RCV5                    ;SCOPE ENTRY POINT         *
                000053                          X=X+1
                                        ;*******************************************************************
                                        ;RECEIVER LINE TEST LINE 5
        012122  004537  005034          RCV5:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 5
        012126  000012                          LINE5
                000006                          Y=Y+1
                                        ;*******************************************************************
        012130  000054                  RT54:   54                      ;ROUTINE # 54              *
        012132  012146                          RT55                    ;ADDRESS OF NEXT TEST      *
        012134  000144                          100.                    ;ITERATION COUNT           *
        012136  012140                          RCV6                    ;SCOPE ENTRY POINT         *
                000054                          X=X+1
                                        ;*******************************************************************
                                        ;RECEIVER LINE TEST LINE 6
        012140  004537  005034          RCV6:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 6
        012144  000014                          LINE6
                000007                          Y=Y+1
                                        ;*******************************************************************
        012146  000055                  RT55:   55                      ;ROUTINE # 55              *
        012150  012164                          RT56                    ;ADDRESS OF NEXT TEST      *
        012152  000144                          100.                    ;ITERATION COUNT           *
        012154  012156                          RCV7                    ;SCOPE ENTRY POINT         *
                000055                          X=X+1
                                        ;*******************************************************************
                                        ;RECEIVER LINE TEST LINE 7
        012156  004537  005034          RCV7:   JSR     5,RCVTST        ;GO TEST RECEIVER LINE 7
        012162  000016                          LINE7
                000010                          Y=Y+1
                                        ;*******************************************************************
        012164  000056                  RT56:   56                      ;ROUTINE # 56              *
        012166  012202                          RT57                    ;ADDRESS OF NEXT TEST      *
        012170  000144                          100.                    ;ITERATION COUNT           *
        012172  012174                          RCV10                   ;SCOPE ENTRY POINT         *
                000056                          X=X+1
                                        ;*******************************************************************
                                        ;RECEIVER LINE TEST LINE 10
        012174  004537  005034          RCV10:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 10
        012200  000020                          LINE10
                000011                          Y=Y+1
                                        ;*******************************************************************
        012202  000057                  RT57:   57                      ;ROUTINE # 57              *
        012204  012220                          RT60                    ;ADDRESS OF NEXT TEST      *
        012206  000144                          100.                    ;ITERATION COUNT           *
        012210  012212                          RCV11                   ;SCOPE ENTRY POINT         *
                000057                          X=X+1
```

```
2447                                    ;*****************************************************************************
2448                                    ;RECEIVER LINE TEST LINE 11
2449  012212  004537  005034           RCV11:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 11
2450  012216  000022                           LINE11
2451          000012                           Y=Y+1
2452                                    ;*****************************************************************************
2453  012220  000060                   RT60:   60                      ;ROUTINE # 60                          *
2454  012222  012236                           RT61                    ;ADDRESS OF NEXT TEST                  *
2455  012224  000144                           100.                    ;ITERATION COUNT                       *
2456  012226  012230                           RCV12                   ;SCOPE ENTRY POINT                     *
2457          000060                           X=X+1
2458                                    ;*****************************************************************************
2459                                    ;RECEIVER LINE TEST LINE 12
2460  012230  004537  005034           RCV12:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 12
2461  012234  000024                           LINE12
2462          000013                           Y=Y+1
2463                                    ;*****************************************************************************
2464  012236  000061                   RT61:   61                      ;ROUTINE # 61                          *
2465  012240  012254                           RT62                    ;ADDRESS OF NEXT TEST                  *
2466  012242  000144                           100.                    ;ITERATION COUNT                       *
2467  012244  012246                           RCV13                   ;SCOPE ENTRY POINT                     *
2468          000061                           X=X+1
2469                                    ;*****************************************************************************
2470                                    ;RECEIVER LINE TEST LINE 13
2471  012246  004537  005034           RCV13:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 13
2472  012252  000026                           LINE13
2473          000014                           Y=Y+1
2474                                    ;*****************************************************************************
2475  012254  000062                   RT62:   62                      ;ROUTINE # 62                          *
2476  012256  012272                           RT63                    ;ADDRESS OF NEXT TEST                  *
2477  012260  000144                           100.                    ;ITERATION COUNT                       *
2478  012262  012264                           RCV14                   ;SCOPE ENTRY POINT                     *
2479          000062                           X=X+1
2480                                    ;*****************************************************************************
2481                                    ;RECEIVER LINE TEST LINE 14
2482  012264  004537  005034           RCV14:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 14
2483  012270  000030                           LINE14
2484          000015                           Y=Y+1
2485                                    ;*****************************************************************************
2486  012272  000063                   RT63:   63                      ;ROUTINE # 63                          *
2487  012274  012310                           RT64                    ;ADDRESS OF NEXT TEST                  *
2488  012276  000144                           100.                    ;ITERATION COUNT                       *
2489  012300  012302                           RCV15                   ;SCOPE ENTRY POINT                     *
2490          000063                           X=X+1
2491                                    ;*****************************************************************************
2492                                    ;RECEIVER LINE TEST LINE 15
2493  012302  004537  005034           RCV15:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 15
2494  012306  000032                           LINE15
2495          000016                           Y=Y+1
2496                                    ;*****************************************************************************
2497  012310  000064                   RT64:   64                      ;ROUTINE # 64                          *
2498  012312  012326                           RT65                    ;ADDRESS OF NEXT TEST                  *
2499  012314  000144                           100.                    ;ITERATION COUNT                       *
2500  012316  012320                           RCV16                   ;SCOPE ENTRY POINT                     *
2501          000064                           X=X+1
2502                                    ;*****************************************************************************
```

```
2503                                    ;RECEIVER LINE TEST LINE 16
2504  012320  004537  005034            RCV16:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 16
2505  012324  000034                            LINE16
2506          000017                            Y=Y+1
2507                                    ;*****************************************************************************
2508  012326  000065                    RT65:   65                      ;ROUTINE # 65                          *
2509  012330  012346                            RT66                    ;ADDRESS OF NEXT TEST                  *
2510  012332  000144                            100.                    ;ITERATION COUNT                       *
2511  012334  012336                            RCV17                   ;SCOPE ENTRY POINT                     *
2512          000065                            X=X+1
2513                                    ;*****************************************************************************
2514                                    ;RECEIVER LINE TEST LINE 17
2515  012336  004537  005034            RCV17:  JSR     5,RCVTST        ;GO TEST RECEIVER LINE 17
2516  012342  000036                            LINE17
2517          000020                            Y=Y+1
2518  012344  000240                            NOP
2519                                    ;*****************************************************************************
2520  012346  000066                    RT66:   66                      ;ROUTINE # 66                          *
2521  012350  012606                            RT67                    ;ADDR OF NEXT ROUTINE.                 *
2522  012352  000012                            10.                     ;ITERATION COUNT                       *
2523  012354  012362                            RT66A                   ;SCOPE ENTRY POINT.                    *
2524          000066                            X=X+1
2525                                    ;*****************************************************************************
2526
2527  012356  000240                            NOP
2528  012360  000240                            NOP
2529                                    ;TEST THAT NEX BIT (CSR BIT 14) SETS WHEN THE TRANSMITTER REFERENCES
2530                                    ;NON-EXISTANT MEMORY. THE CORRESPONDING BAR BIT CLEARS
2531                                    ;AND THAT AN INTERRUPT OCCURS. ALL LINES ARE USED FOR THE TEST
2532  012362  004737  004216            RT66A:  JSR     7,TIMER         ;GO CALCULATE MACINE TIME TO TRANSMIT
2533                                                                    ;ONE CHARACTER
2534  012366  012701  001400                    MOV     #LAT,X1         ;GET CAT ADDRESS
2535  012372  012702  160000                    MOV     #160000,X2      ;GET A NON-EXISTANT ADDRESS
2536  012376  012703  000020                    MOV     #16.,X3         ;GET COUNTER
2537  012402  010221                    1$      MOV     X2,(1)+         ;LOAD THE CURRENT ADDRESS
2538  012404  005303                            DEC     X3              ;TABLE WITH NON-EXISTANT
2539  012406  001375                            BNE     1$              ;ADDRESSES
2540  012410  012701  000001                    MOV     #LBIT0,X1       ;GET LINE BIT
2541  012414  012777  012552  167134            MOV     #6$,@XMTINT     ;LOAD TRANSMITTER INT. VECTOR
2542  012422  052777  000060  167112  2$:       BIS     #60,@CSR        ;SET EXTENDED ADDRESS BITS
2543  012430  050177  167110                    BIS     X1,@BAR         ;START TRANSMITTER
2544  012434  013737  004370  012444            MOV     TIME14,3$       ;LOAD DELAY TIME TO
2545  012442  104400                            DELAY                   ;DELAY FOR 1/4TH OF A CHARACTER
2546  012444  000000                    3$:     OPEN                    ;TO RESPOND TO NEX
2547  012446  017737  167072  001566            MOV     @BAR,RCVDAT     ;GET BAR DATA & TEST
2548  012454  001406                            BEQ     4$              ;THAT IT IS CLEAR
2549  012456  005037  001570                    CLR     XMTDAT
2550  012462  104011                            ERROR1                  ;ERROR!BAR BIT DID NOT CLEAR
2551  012464  005077  167054                    CLR     @BAR
2552  012470  000440                            BR      7$              ;GO TO SCOPE
2553  012472  032777  040000  167042  4$:       BIT     #BIT14,@CSR     ;TEST THAT NEX BIT IS SET
2554  012500  001002                            BNE     5$              ;BRANCH IF SET
2555  012502  104001                            ERROR                   ;ERROR! NEX BIT FAILED TO SET
2556  012504  000432                            BR      7$              ;GO TO SCOPE
2557  012506  042777  100000  167026  5$:       BIC     #BIT15,@CSR     ;CLEAR TRANSMITTER READY FLAG
2558  012514  052777  010000  167020            BIS     #BIT12,@CSR     ;SET TRANSMITTER IE BIT
```

```
2559  012522  005037  177776            CLR     @#PSW           ;ALLOW INTERRUPTS
2560  012526  000240                    NOP
2561  012530  012737  000340  177776    MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
2562  012536  010137  001570            MOV     X1,XMTDAT       ;LOAD LINE THAT FAILED
2563  012542  005037  001566            CLR     RCVDAT
2564  012546  104011                    ERROR1                  ;ERROR! NEX FAILED TO CAUSE INTERRUPT
2565                                                            ;TYPEOUT SHOWS LINE # THAT FAILED
2566  012550  000410                    BR      7$              ;GO TO SCOPE
2567  012552  005077  166764    6$:     CLR     @CSR
2568  012556  012737  000340  177776    MOV     #PRTY7,@#PSW    ;LOCK OUT INTERRUPTS
2569  012564  022626                    CMP     (6)+,(6)+       ;ADJUST STACK PTR
2570  012566  006301                    ASL     X1              ;SHIFT LINE BIT
2571  012570  103314                    BCC     2$              ;DO NEXT LINE
2572  012572  013737  004366  012602  7$:     MOV     TIME1,8$        ;WAIT FOR TRANSMITTER TO RUN
2573  012600  104400                    DELAY                   ;TO COMPLETION BEFORE
2574  012602  000000            8$:     OPEN                    ;EXITING TEST
2575  012604  104006                    SCOPE                   ;SCOPE
2576                            ;*****************************************************************
2577  012606  000067            RT67:   67                      ;ROUTINE # 67                   *
2578  012610  012716                    RT70                    ;ADDR OF NEXT ROUTINE           *
2579  012612  000012                    10.                     ;ITERATION COUNT                *
2580  012614  012616                    RT67A                   ;SCOPE ENTRY POINT.             *
2581          000067                    X=X+1
2582                            ;*****************************************************************
2583
2584                            ;TEST THAT NEX BIT SETS IF THE DM11 TABLES ARE IN NON-EXISTANT CORE
2585  012616  012777  160000  166724  RT67A:  MOV     #160000,@#ASREG ;SET BASE REGISTER TO NON-EXISTANT ADRS
2586  012624  012737  012706  000004          MOV     #4$,@#ERRVEC    ;SET TIME OUT TRAP VECTOR
2587  012632  005737  160000                  TST     @#160000        ;CHECK THAT ADDRESS TIMES OUT
2588  012636  013737  004370  012654          MOV     TIME1/4,1$      ;GET TIME TO TRANSMIT 1/4 CHAR
2589  012644  052777  000001  166672          BIS     #LBIT0,@#AR     ;START TO TRANSMIT ON LINE 0
2590  012652  104400                          DELAY                   ;DELAY 1/4TH OF A CHARACTER
2591  012654  000000            1$:     OPEN                    ;TIME
2592  012656  005077  166662                  CLR     @#AR            ;STOP TRANSMITTER
2593  012662  022777  040060  166652          CMP     #BIT14+60,@#CSR ;TEST THAT ONLY NEX IS SET
2594  012670  001401                          BEQ     2$
2595  012672  104001                          ERROR                   ;ERROR! EITHER NEX FAILED TO SET
2596                                                                  ;OR OTHER BITS SET
2597  012674  013737  004366  012704  2$:     MOV     TIME1,3$        ;DELAY 1 CHARACTER TIME TO ALLOW
2598  012702  104400                          DELAY                   ;TRANSMITTER TO RUN TO
2599  012704  000000            3$:     OPEN                    ;COMPLETION
2600  012706  012737  000006  000004  4$:     MOV     @ERRVEC+2,@#ERRVEC      ;RESTORE TIME OUT TRAP
2601  012714  104006                          SCOPE
2602                            ;*****************************************************************
2603  012716  000070            RT70:   70                      ;ROUTINE # 70                   *
2604  012720  013032                    RT71                    ;ADDR OF NEXT ROUTINE.          *
2605  012722  000144                    100.                    ;ITERATION COUNT                *
2606  012724  012726                    RT70A                   ;SCOPE ENTRY POINT.             *
2607          000070                    X=X+1
2608                            ;*****************************************************************
2609
2610                            ;TEST THAT WHEN THE GO BIT IS CLEAR THAT THE RECEIVERS DO NOT RECEIVE
2611                            ;DATA. EACH LINE IN TURN IS TRANSMITTED ON, AND WHEN TEN CHARACTERS
2612                            ;HAVE BEEN TRANSMITTED THE RECEIVER DONE FLAG IS TESTED.  IF IT IS SET
2613                            ;AN ERROR IS INDICATED ON THE LINE DATA WAS RECEIVED ON.
2614  012726  005037  016730    RT70A:  CLR     LINE            ;SET UP TO TRANSMIT
```

```
 2615  012732  004537  006246          1$.     JSR     5,@#XMITD       ,10  CHARACTERS
 2616  012736  177766                          -10.                    ,ON EACH LINE
 2617  012740  005777  166576                  TST     @CSR            ,WAIT FOR 10  CHARACTERS
 2618  012744  100375                          BPL     .-4             ,TO BE TRANSMITTED
 2619  012746  042777  100000  166566          BIC     #100000,@CSR
 2620  012754  105777  166562                  TSTB    @CSR            ;TEST RECEIVER DONE FLAG
 2621  012760  100010                          BPL     2$
 2622  012762  013737  001564  001566          MOV     LINBIT,RCVDAT   ,GET LINE BIT OF ACTIVE LINE
 2623  012770  013737  001564  001570          MOV     LINBIT,XMTDAT   ,THAT ERROR OCCURED ON
 2624  012776  104011                          ERROR1                  ,ERROR! DATA WAS RECEIVED ON LINE INDICATED
 2625  013000  000413                          BR      4$              ,GO TO SCOPE
 2626  013002  062737  000002  016730  2$:     ADD     #2,LINE         ,SET UP NEXT LINE NUMBER
 2627  013010  006337  001564                  ASL     LINBIT          ,GET READY TO TRANSMIT ON NEXT LINE
 2628  013014  103346                          BCC     1$              ,GO TRANSMIT ON NEXT LINE
 2629  013016  013737  004366  013026          MOV     TIME1,3$
 2630  013024  104400                          DELAY                   ;DELAY 1 CHARACTER
 2631  013026  000000                  3$:     0                       ;TIME BEFORE ENTERING NEXT TEST
 2632  013030  104006                  4$:     SCOPE                   ;SCOPE
 2633                                  ;*************************************************************************
 2634  013032  000071                  RT71:   71                      ;ROUTINE # 71                          *
 2635  013034  013226                          RT72                    ;ADDR OF NEXT ROUTINE                  *
 2636  013036  000024                          20.                     ;ITERATION COUNT                       *
 2637  013040  013042                          RT71A                   ;SCOPE ENTRY POINT.                    *
 2638          000071                          X=X+1
 2639                                  ;*************************************************************************
 2640
 2641                                  ;TEST THAT CURRENT ADDRESS INCREMENTS PROPERLY WHEN A CHAR-
 2642                                  ;ACTER IS TRANSMITTED.  LINE 0 IS USED FOR THE TEST.
 2643  013042  005000                  RT71A:  CLR     X0              ;R0=CURRENT ADRS AFTER TRANSMISSION
 2644  013044  010001                  1$:     MOV     X0,X1           ;R0=CURRENT ADDRESS BEFORE TRANSMISSION
 2645  013046  005201                          INC     X1              ;AND R1=CURRENT ADDRESS AFTER TRANSMISSION
 2646  013050  012737  013212  000004          MOV     #3$,@#ERRVEC            ;SET UP PROCESSOR
 2647  013056  012737  000340  000006          MOV     #PRTY7,@#ERRVEC+2       ;TIME OUT TRAP
 2648  013064  012777  001400  166456          MOV     #CAT,@#BAREG    ;SET UP BASE REGISTER
 2649  013072  010037  001400                  MOV     X0,CAT          ;LOAD CURRENT ADDRESS TABLE (LINE 0)
 2650  013076  105710                          TSTB    (0)             ;DOES MEMORY EXIST?
 2651  013100  012737  177776  001440          MOV     #-2,WCT         ;SET CHAR. COUNT TO TRANSMIT 1 CHAR.
 2652  013106  012777  000006  166426          MOV     #6,@CSR         ;SET MAINT & GO BITS
 2653  013114  012777  000001  166422          MOV     #LBIT0,@BAR     ;TRANSMIT ON LINE 0
 2654  013122  105777  166414                  TSTB    @CSR            ;WAIT FOR THE RECEIVER
 2655  013126  100375                          BPL     .-4             ;TO RECEIVE FIRST CHARACTER
 2656  013130  042777  000200  166404          BIC     #200,@CSR       ;CLEAR RECEIVER DONE FLAG
 2657  013136  105777  166400                  TSTB    @CSR            ;WAIT FOR RECEIVER TO RECEIVE
 2658  013142  100375                          BPL     .-4             ;THE SECOND CHARACTER
 2659  013144  023701  001400                  CMP     CAT,X1          ;TEST THAT CURRENT ADRS
 2660                                                                  ;INCREMENTED PROPERLY
 2661  013150  001413                          BEQ     2$
 2662  013152  010137  001570                  MOV     X1,XMTDAT       ;GET COMPUTED RESULT
 2663  013156  013737  001400  001566          MOV     CAT,RCVDAT      ;GET ACTUAL RESULT
 2664  013164  104011                          ERROR1                  ;ERROR! CURRENT ADDRESS DID NOT
 2665  013166  032777  040000  166006          BIT     #BIT14,@SWR     ;INCREMENT PROPERLY
 2666  013174  001323                          BNE     1$              ;BRANCH IF SCOPE SWITCH IS SET
 2667  013176  000405                          BR      3$              ;GO TO EXIT
 2668  013200  005701                  2$:     TST     X1
 2669  013202  001403                          BEQ     3$
 2670  013204  000261                          SEC
```

```
2671  013206  006100                          ROL     %0
2672  013210  100715                          BMI     1$
2673  013212  012737  000006  000004  3$:     MOV     #ERRVEC+2,@#ERRVEC      ;RESTORE TIME OUT TRAP
2674  013220  005037  000006                  CLR     @#ERRVEC+2
2675  013224  104006                          SCOPE                           ;SCOPE
2676                                ;**************************************************************************
2677  013226  000072                RT72:     72                              ;ROUTINE # 72                    *
2678  013230  013572                          RT73                            ;ADDR OF NEXT ROUTINE.           *
2679  013232  000024                          20.                             ;ITERATION COUNT                 *
2680  013234  013236                          RT72A                           ;SCOPE ENTRY POINT.              *
2681           000072                         X=X+1
2682                                ;**************************************************************************
2683
2684                                ;TEST THAT DATA CAN BE TRANSMITTED FROM ALL AVAILABLE CORE.
2685                                ;LINE 0 IS USED FOR THE TEST AND ONLY ONE WORD IS TRANSMITTED
2686                                ;AT A TIME.
2687  013236  005000                RT72A:    CLR     %0                      ;CLEAR INDEX REGISTER
2688  013240  000005                          RESET
2689  013242  016037  013526  001400  1$:     MOV     AREA(0),CAT             ;LOAD CURRENT ADDRESS
2690  013250  012777  000005  166264          MOV     #5,@CSR                 ;SET MAINT & GO BITS
2691  013256  012737  177777  001440          MOV     #-1,WCT                 ;SET UP CHAR COUNT TO TRANSMIT 1 CHAR
2692  013264  012777  000001  166252          MOV     #LBIT0,@BAR             ;TRANSMIT CHAR ON LINE 0
2693  013272  005777  166244                  TST     @CSR                    ;WAIT FOR THE TRANSMITTER
2694  013276  100375                          BPL     .-4                     ;TO TRANSMIT THE CHARACTER
2695  013300  105777  166236                  TSTB    @CSR                    ;TEST FOR DONE
2696  013304  100375                          BPL     .-4
2697  013306  005077  166230                  CLR     @CSR                    ;CLEAR ALL FLAGS
2698  013312  005037  001566                  CLR     RCVDAT
2699  013316  113737  001600  001566          MOVB    TUMTAB,RCVDAT           ;GET RECEIVED CHARACTER
2700  013324  117037  013526  001570          MOVB    @AREA(0),XMTDAT         ;GET TRANSMITTED CHARACTER
2701  013332  043737  001572  001570          BIC     CARMSK,XMTDAT           ;CLEAR NON-TRANSMITTED BITS
2702  013340  123737  001566  001570          CMPB    RCVDAT,XMTDAT           ;COMPARE CHARACTERS
2703  013346  001402                          BEQ     2$                      ;BRANCH IF VALID COMPARISON
2704  013350  104011                          ERROR1                          ;ERROR!DATA COMPARISON ERROR
2705                                ;(CAT)-1 IS THE MEMORY LOCATION WHERE THE DATA WAS TRANSMITTED FROM
2706  013352  000464                          BR      6$                      ;GO TO EXIT
2707  013354  020027  000006        2$:     CMP     %0,#6                   ;HAS FIRST 4K BEEN TESTED
2708  013360  001402                          BEQ     3$                      ;BRANCH IF IT HAS
2709  013362  005720                          TST     (0)+                    ;INCREMENT INDEX
2710  013364  000726                          BR      1$                      ;GO REPEAT TEST
2711
2712
2713  013366                        3$:                                       ;BEGIN TESTING ABOVE 4K
2714  013366  012737  013514  000004          MOV     #5$,@#ERRVEC            ;SET TIME OUT TRAP TO EXIT
2715                                                                          ;TEST IF MEMORY TIMES OUT
2716  013374  005001                          CLR     %1                      ;SET UP DATA IDENTIFIER
2717  013376  005201                4$:     INC     %1                      ;INCREMENT DATA IDENTIFIER
2718  013400  005720                          TST     (0)+                    ;INCREMENT INDEX
2719  013402  110170  013526                  MOVB    %1,@AREA(0)             ;LOAD IDENTIFIER INTO MEMORY
2720  013406  016037  013526  001400          MOV     AREA(0),CAT             ;LOAD CURRENT ADDRESS
2721  013414  012777  000005  166120          MOV     #5,@CSR                 ;SET MAINT & GO BITS
2722  013422  012737  177777  001440          MOV     #-1,WCT                 ;SET UP CHAR COUNT TO TRANSMIT 1 CHAR
2723  013430  012777  000001  166106          MOV     #LBIT0,@BAR             ;TRANSMIT ON LINE 0
2724  013436  005777  166100                  TST     @CSR                    ;WAIT FOR THE TRANSMITTER TO
2725  013442  100375                          BPL     .-4                     ;TRANSMIT THE CHARACTER
2726  013444  105777  166072                  TSTB    @CSR                    ;TEST FOR CHARACTER DONE
```

```
2727  013450  100375                          BPL     .-4
2728  013452  005077  166064                  CLR     @CSR
2729  013456  113737  001600  001566          MOVB    TUMTAB,RCVDAT   ;GET THE RECEIVED CHARACTER
2730  013464  117037  013526  001570          MOVB    @AREA(0),XMTDAT ;GET THE TRANSMITTED CHARACTER
2731  013472  043737  001572  001570          BIC     CARMSK,XMTDAT   ;CLEAR NON-TRANSMITTED BITS
2732  013500  123737  001566  001570          CMPB    RCVDAT,XMTDAT   ;COMPARE CHARACTERS
2733  013506  001733                          BEQ     4$              ;BRANCH IF VALID COMPARISON
2734  013510  104011                          ERROR1                  ;ERROR!DATA COMPARISON ERROR NUMBER
2735  013512  000404                          BR      6$              ;IN S/B GIVES MEMORY LOCATION (SEE TABLE
2736  013514  022626                  5$:     POPSP2                  ;RESET THE STACK
2737  013516  012737  000006  000004          MOV     #6,@#ERRVEC     ;RESTORE TIME OUT TRAP
2738  013524  104006                  4$:     SCOPE                   ;EXIT TEST
2739                                          ;MEMORY LOCATIONS TRANSMITTED FROM TABLE
2740  013526  000000                  AREA:   0                       ;FOR DATA IN FIRST
2741  013530  005252                          5252                    ;4K SEE THE LISTING
2742  013532  012525                          12525                   ;CONTENTS OF THESE LOCATIONS (BYTE)
2743  013534  017777                          17777                   ;IS THE DATA TRANSMITTED
2744  013536  020000                  A8K.    20000                   ;CONTENTS =1 (IF AVAILABLE)
2745  013540  026314                          26314                   ;    "     2       "
2746  013542  031463                          31463                   ;    "     3       "
2747
2748  013544  037477                          37477                   ;    "     4       "
2749  013546  040000                  A12K:   40000                   ;    "     5       "
2750  013550  057477                          57477                   ;    "     6       "
2751  013552  060000                  A16K:   60000                   ;    "     7       "
2752  013554  077477                          77477                   ;    "     10      "
2753  013556  100000                  A20K:   100000                  ;    "     11      "
2754  013560  117477                          117477                  ;    "     12      "
2755  013562  120000                  A24K:   120000                  ;    "     13      "
2756  013564  137477                          137477                  ;    "     14      "
2757  013566  140000                  A28K:   140000                  ;    "     15      "
2758  013570  173000                          173000                  ;    "     16      "
2759                                          ;*************************************************************************
2760  013572  000073                  RT73:   73                      ;ROUTINE # 73                                   *
2761  013574  014006                          RT74                    ;ADDR OF NEXT ROUTINE.                          *
2762  013576  000012                          10.                     ;ITERATION COUNT                                *
2763  013600  013602                          RT73A                   ;SCOPE ENTRY POINT.                             *
2764          000073                          X=X+1
2765                                          ;*************************************************************************
2766
2767                                          ;TEST THAT THE TRANSMITTER CAN TRANSMIT 100. CHARACTERS BEFORE SETTING
2768                                          ;THE READY BIT (CSR 15),AND CLEARING THE BAR BIT
2769  013602  005037  016730          RT73A:  CLR     LINE
2770  013606  012777  000001  165726  1$:     MOV     #1,@CSR         ;SET THE GO BIT
2771  013614  005037  001566                  CLR     RCVDAT
2772  013620  013737  016730  001570          MOV     @#LINE,XMTDAT   ;GET LINE NUMBER (X2)
2773  013626  004537  006246                  JSR     5,@#XMITD       ;TRANSMIT 100. CHARACTERS
2774  013632  177634                          -100.                   ;ON LINE AS SPECIFIED BY LINE
2775  013634  106777  165702          2$:     TSTB    @CSR            ;WAIT FOR THE RECEIVER
2776  013640  100375                          BPL     2$              ;TO RECEIVE ONE CHARACTER
2777  013642  042777  000200  165672          BIC     #BIT7,@CSR      ;CLEAR CHAR. DONE FLAG
2778  013650  005237  001566                  INC     RCVDAT          ;INCREMENT CHAR. RCVD COUNT
2779  013654  023727  001566  000144          CMP     RCVDAT,#100     ;HAVE 100. CHARS. BEEN RCVD
2780  013662  001416                          BEQ     4$
2781  013664  005777  165652                  TST     @CSR            ;TEST READY FLAG
2782  013670  100002                          BPL     3$              ;GO TEST BAR
```

```
2783  013672  104011                          ERROR1                  ;ERROR!READY BIT SET TOO SOON
2784                              ;TYPEOUT SHOWS HOW MANY CHARS WERE RECEIVED WHEN READY SET AND THE LINE # (X2)
2785  013674  000443                          BR      8$              ;GO TO EXIT
2786
2787  013676  023777  001564  165640  3$:     CMP     @#LINBIT,@BAR   ;TEST THAT BAR BIT IS SET
2788  013704  001753                          BEQ     2$              ;BRANCH IF SET
2789  013706  017737  165632  001570          MOV     @BAR,XMTDAT     ;GET BAR CONTENTS
2790  013714  104011                          ERROR1                  ;ERROR! BAR BIT CLEARED TO SOON
2791                              ;TYPEOUT SHOWS THE BAR CONTENTS AND HOW MANY CHARS WERE RECEIVED WHEN BAR FAILED
2792                              ;LOCATION LINBIT HAS THE CORRECT BAR CONTENTS.
2793  013716  000432                          BR      8$              ;EXIT TEST
2794  013720  013737  004370  013730  4$:     MOV     TIME14,5$       ;DELAY 1/4 CHARACTER TIME
2795  013726  104400                          DELAY                   ;TO ALLOW TRANSMITTER TO FINISH
2796  013730  000000                  5$:     OPEN
2797  013732  005777  165604                  TST     @CSR            ;TEST READY FLAG (SHOULD BE SET)
2798  013736  100402                          BMI     6$              ;GO TEST BAR
2799  013740  104001                          ERROR                   ;ERROR! READY FLAG FAILED TO SET
2800  013742  000420                          BR      8$              ;GO TO EXIT
2801  013744  005777  165574          6$:     TST     @BAR            ;TEST THAT BAR BIT IS CLEAR
2802  013750  001407                          BEQ     7$              ;GO TO 7$ IF CLEAR
2803  013752  017737  165566  001566          MOV     @BAR,RCVDAT
2804  013760  005037  001570                  CLR     XMTDAT
2805  013764  104011                          ERROR1                  ;ERROR! BAR BIT FAILED TO CLEAR
2806  013766  000406                          BR      8$
2807  013770  062737  000002  016730  7$:     ADD     #2,@#LINE
2808  013776  006337  001564                  ASL     @#LINBIT
2809  014002  103301                          BCC     1$
2810  014004  104006                  8$:     SCOPE
2811                              ;*************************************************************************
2812  014006  000074                  RT74:   74                      ;ROUTINE # 74                    *
2813  014010  014174                          RT75                    ;ADDR OF NEXT ROUTINE            *
2814  014012  000012                          10.                     ;ITERATION COUNT                 *
2815  014014  014016                          RT74A                   ;SCOPE ENTRY POINT               *
2816          000074                          X=X+1
2817                              ;*************************************************************************
2818
2819                              ;TEST THAT THE TUMBLE TABLE POINTER INCREMENTS PROPERLY AND
2820                              ;RETURNS TO THE BEGINNING AFTER 64. CHARACTERS HAVE BEEN RECEIVED
2821                              ;LINE 0 IS USED FOR THE TEST
2822  014016  012701  001600  RT74A:  MOV     #TUMTAB,X1      ;CLEAR THE
2823  014022  012702  000100          MOV     #64.,X2         ;TUMBLE TABLE
2824  014026  005021          1$:     CLR     (1)+
2825  014030  005302                  DEC     X2
2826  014032  001375                  BNE     1$
2827  014034  012701  001600          MOV     #TUMTAB,X1
2828  014040  012777  000004  165474  MOV     #BIT2,@CSR      ;SET MAINT BIT & CLEAR GO BIT
2829  014046  005037  001570          CLR     XMTDAT
2830  014052  005037  001566          CLR     RCVDAT
2831  014056  012737  177677  001440  MOV     #-65.,WCT       ;SET UP TO TRANSMIT 65  CHARACTERS
2832  014064  052777  000001  165450  BIS     #BIT0,@CSR      ;SET THE GO BIT
2833  014072  012777  000001  165444  MOV     #LBIT0,@BAR     ;TRANSMIT ON LINE 0
2834  014100  105777  165436          2$:     TSTB    @CSR            ;WAIT FOR CHAR  DONE FLPG
2835  014104  100375                  BPL     2$
2836  014106  042777  000200  165426  BIC     #BIT7,  @CSR    ;CLEAR CHAR  DONE FLAG
2837  014114  005237  001566          INC     RCVDAT          ;INCREMENT CHARACTERS
2838  014120  005237  001570          INC     XMTDAT          ;RECEIVED COUNT
```

```
2839  014124  005711                          TST     (1)             ;TEST TT ENTRY FOR VALID
2840  014126  100402                          BMI     3$              ;DATA ENTRY
2841  014130  104011                          ERROR1                  ;ERROR! NO VALID DATA ENTRY
2842                                          ;TYPEOUT SHOWS # OF CHARS  RCVD WHEN ERROR OCCURED
2843  014132  000417                          BR      4$              ;GO TO SCOPE
2844  014134  005021                  3$:     CLR     (1)+            ;CLEAR TT ENTRY
2845  014136  023727  001570  000100         CMP     XMTDAT,#64.     ;HAVE 64. CHARACTERS BEEN RECEIVED
2846  014144  001355                          BNE     2$
2847  014146  005777  165370                  TST     @CSR            ;WAIT FOR THE LAST CHARACTER
2848  014152  100375                          BPL     .-4             ;TO BE TRANSMITTED
2849  014154  105777  165362                  TSTB    @CSR            ;TEST FOR DONE
2850  014160  100375                          BPL     .-4
2851  014162  005737  001600                  TST     TUMTAB          ;TEST FIRST TT ENTRY
2852  014166  100401                          BMI     4$              ;FOR VALID DATA
2853  014170  104001                          ERROR                   ;ERROR! POINTER DID NOT RETURN
2854  014172  104006                  4$:     SCOPE                   ;SCOPE
2855          000000                          A=0
2856          000000                          Y=0
2857                                  ;*******************************************************************************
2858  014174  000075                  RT75:   75                      ;ROUTINE # 75                                  *
2859  014176  014212                          RT76                    ;ADDRESS OF NEXT TEST                          *
2860  014200  000144                          100.                    ;ITERATION COUNT                               *
2861  014202  014204                          BRK0                    ;SCOPE ENTRY POINT                             *
2862          000075                          X=X+1
2863                                  ;*******************************************************************************
2864                                  ;BREAK TEST ON LINE 0.
2865  014204  004537  005456          BRK0:   JSR     5,BRKTST        ;GO DO BREAK TEST
2866  014210  000001                          LBIT0                   ;ON LINE 0
2867          000001                          Y=Y+1
2868                                  ;*******************************************************************************
2869  014212  000076                  RT76:   76                      ;ROUTINE # 76                                  *
2870  014214  014230                          RT77                    ;ADDRESS OF NEXT TEST                          *
2871  014216  000144                          100.                    ;ITERATION COUNT                               *
2872  014220  014222                          BRK1                    ;SCOPE ENTRY POINT                             *
2873          000076                          X=X+1
2874                                  ;*******************************************************************************
2875                                  ;BREAK TEST ON LINE 1.
2876  014222  004537  005456          BRK1:   JSR     5,BRKTST        ;GO DO BREAK TEST
2877  014226  000002                          LBIT1                   ;ON LINE 1
2878          000002                          Y=Y+1
2879                                  ;*******************************************************************************
2880  014230  000077                  RT77:   77                      ;ROUTINE # 77                                  *
2881  014232  014246                          RT100                   ;ADDRESS OF NEXT TEST                          *
2882  014234  000144                          100.                    ;ITERATION COUNT                               *
2883  014236  014240                          BRK2                    ;SCOPE ENTRY POINT                             *
2884          000077                          X=X+1
2885                                  ;*******************************************************************************
2886                                  ;BREAK TEST ON LINE 2
2887  014240  004537  005456          BRK2:   JSR     5,BRKTST        ;GO DO BREAK TEST
2888  014244  000004                          LBIT2                   ;ON LINE 2
2889          000003                          Y=Y+1
2890                                  ;*******************************************************************************
2891  014246  000100                  RT100:  100                     ;ROUTINE # 100                                 *
2892  014250  014264                          RT101                   ;ADDRESS OF NEXT TEST                          *
2893  014252  000144                          100.                    ;ITERATION COUNT                               *
2894  014254  014256                          BRK3                    ;SCOPE ENTRY POINT                             *
```

```
                000100                  X=X+1
                                ;*********************************************************************
                                ;BREAK TEST ON LINE 3.
        014256  004537  005456  BRK3:   JSR     5,BRKTST                ;GO DO BREAK TEST
        014262  000010                  LBIT3                           ;ON LINE 3
                000004                  Y=Y+1
                                ;*********************************************************************
        014264  000101          RT101:  101                             ;ROUTINE # 101              *
        014266  014302                  RT102                           ;ADDRESS OF NEXT TEST       *
        014270  000144                  100.                            ;ITERATION COUNT            *
        014272  014274                  BRK4                            ;SCOPE ENTRY POINT          *
                000101                  X=X+1
                                ;*********************************************************************
                                ;BREAK TEST ON LINE 4.
        014274  004537  005456  BRK4:   JSR     5,BRKTST                ;GO DO BREAK TEST
        014300  000020                  LBIT4                           ;ON LINE 4
                000005                  Y=Y+1
                                ;*********************************************************************
        014302  000102          RT102:  102                             ;ROUTINE # 102              *
        014304  014320                  RT103                           ;ADDRESS OF NEXT TEST       *
        014306  000144                  100.                            ;ITERATION COUNT            *
        014310  014312                  BRK5                            ;SCOPE ENTRY POINT          *
                000102                  X=X+1
                                ;*********************************************************************
                                ;BREAK TEST ON LINE 5.
        014312  004537  005456  BRK5:   JSR     5,BRKTST                ;GO DO BREAK TEST
        014316  000040                  LBIT5                           ;ON LINE 5
                000006                  Y=Y+1
                                ;*********************************************************************
        014320  000103          RT103:  103                             ;ROUTINE # 103              *
        014322  014336                  RT104                           ;ADDRESS OF NEXT TEST       *
        014324  000144                  100.                            ;ITERATION COUNT            *
        014326  014330                  BRK6                            ;SCOPE ENTRY POINT          *
                000103                  X=X+1
                                ;*********************************************************************
                                ;BREAK TEST ON LINE 6.
        014330  004537  005456  BRK6:   JSR     5,BRKTST                ;GO DO BREAK TEST
        014334  000100                  LBIT6                           ;ON LINE 6
                000007                  Y=Y+1
                                ;*********************************************************************
        014336  000104          RT104:  104                             ;ROUTINE # 104              *
        014340  014354                  RT105                           ;ADDRESS OF NEXT TEST       *
        014342  000144                  100.                            ;ITERATION COUNT            *
        014344  014346                  BRK7                            ;SCOPE ENTRY POINT          *
                000104                  X=X+1
                                ;*********************************************************************
                                ;BREAK TEST ON LINE 7.
        014346  004537  005456  BRK7:   JSR     5,BRKTST                ;GO DO BREAK TEST
        014352  000200                  LBIT7                           ;ON LINE 7
                000010                  Y=Y+1
                                ;*********************************************************************
        014354  000105          RT105:  105                             ;ROUTINE # 105              *
        014356  014372                  RT106                           ;ADDRESS OF NEXT TEST       *
        014360  000144                  100.                            ;ITERATION COUNT            *
        014362  014364                  BRK10                           ;SCOPE ENTRY POINT          *
                000105                  X=X+1
```

```
2951                                    ;*****************************************************************************
2952                                    ;BREAK TEST ON LINE 10.
2953  014364  004537  005456             BRK10:  JSR     5,BRKTST        ;GO DO BREAK TEST
2954  014370  000400                             LBIT10                  ;ON LINE 10
2955           000011                             Y=Y+1
2956                                    ;*****************************************************************************
2957  014372  000106                     RT106:  106                     ;ROUTINE # 106                          *
2958  014374  014410                             RT107                   ;ADDRESS OF NEXT TEST                   *
2959  014376  000144                             100.                    ;ITERATION COUNT                        *
2960  014400  014402                             BRK11                   ;SCOPE ENTRY POINT                      *
2961           000106                             X=X+1
2962                                    ;*****************************************************************************
2963                                    ;BREAK TEST ON LINE 11.
2964  014402  004537  005456             BRK11:  JSR     5,BRKTST        ;GO DO BREAK TEST
2965  014406  001000                             LBIT11                  ;ON LINE 11
2966           000012                             Y=Y+1
2967                                    ;*****************************************************************************
2968  014410  000107                     RT107:  107                     ;ROUTINE # 107                          *
2969  014412  014426                             RT110                   ;ADDRESS OF NEXT TEST                   *
2970  014414  000144                             100.                    ;ITERATION COUNT                        *
2971  014416  014420                             BRK12                   ;SCOPE ENTRY POINT                      *
2972           000107                             X=X+1
2973                                    ;*****************************************************************************
2974                                    ;BREAK TEST ON LINE 12.
2975  014420  004537  005456             BRK12:  JSR     5,BRKTST        ;GO DO BREAK TEST
2976  014424  002000                             LBIT12                  ;ON LINE 12
2977           000013                             Y=Y+1
2978                                    ;*****************************************************************************
2979  014426  000110                     RT110:  110                     ;ROUTINE # 110                          *
2980  014430  014444                             RT111                   ;ADDRESS OF NEXT TEST                   *
2981  014432  000144                             100.                    ;ITERATION COUNT                        *
2982  014434  014436                             BRK13                   ;SCOPE ENTRY POINT                      *
2983           000110                             X=X+1
2984                                    ;*****************************************************************************
2985                                    ;BREAK TEST ON LINE 13.
2986  014436  004537  005456             BRK13:  JSR     5,BRKTST        ;GO DO BREAK TEST
2987  014442  004000                             LBIT13                  ;ON LINE 13
2988           000014                             Y=Y+1
2989                                    ;*****************************************************************************
2990  014444  000111                     RT111:  111                     ;ROUTINE # 111                          *
2991  014446  014462                             RT112                   ;ADDRESS OF NEXT TEST                   *
2992  014450  000144                             100.                    ;ITERATION COUNT                        *
2993  014452  014454                             BRK14                   ;SCOPE ENTRY POINT                      *
2994           000111                             X=X+1
2995                                    ;*****************************************************************************
2996                                    ;BREAK TEST ON LINE 14.
2997  014454  004537  005456             BRK14:  JSR     5,BRKTST        ;GO DO BREAK TEST
2998  014460  010000                             LBIT14                  ;ON LINE 14
2999           000015                             Y=Y+1
3000                                    ;*****************************************************************************
3001  014462  000112                     RT112:  112                     ;ROUTINE # 112                          *
3002  014464  014500                             RT113                   ;ADDRESS OF NEXT TEST                   *
3003  014466  000144                             100.                    ;ITERATION COUNT                        *
3004  014470  014472                             BRK15                   ;SCOPE ENTRY POINT                      *
3005           000112                             X=X+1
3006                                    ;*****************************************************************************
```

```
3007                                    ;BREAK TEST ON LINE 15
3008  014472  004537  005456            BRK15:  JSR     5,BRKTST        ;GO DO BREAK TEST
3009  014476  020000                            LBIT15                  ;ON LINE 15
3010          000016                            Y=Y+1
3011                                    ;*****************************************************************************
3012  014500  000113                    RT113:  113                     ;ROUTINE # 113                          *
3013  014502  014516                            RT114                   ;ADDRESS OF NEXT TEST                   *
3014  014504  000144                            100.                    ;ITERATION COUNT                        *
3015  014506  014510                            BRK16                   ;SCOPE ENTRY POINT                      *
3016          000113                            X=X+1
3017                                    ;*****************************************************************************
3018                                    ;BREAK TEST ON LINE 16.
3019  014510  004537  005456            BRK16:  JSR     5,BRKTST        ;GO DO BREAK TEST
3020  014514  040000                            LBIT16                  ;ON LINE 16
3021          000017                            Y=Y+1
3022                                    ;*****************************************************************************
3023  014516  000114                    RT114.  114                     ;ROUTINE # 114                          *
3024  014520  014534                            RT115                   ;ADDRESS OF NEXT TEST                   *
3025  014522  000144                            100.                    ;ITERATION COUNT                        *
3026  014524  014526                            BRK17                   ;SCOPE ENTRY POINT                      *
3027          000114                            X=X+1
3028                                    ;*****************************************************************************
3029                                    ;BREAK TEST ON LINE 17.
3030  014526  004537  005456            BRK17:  JSR     5,BRKTST        ;GO DO BREAK TEST
3031  014532  100000                            LBIT17                  ;ON LINE 17
3032          000020                            Y=Y+1
3033          000000                            A=0
3034          000000                            Y=0
3035                                    ;*****************************************************************************
3036  014534  000115                    RT115:  115                     ;ROUTINE #115                           *
3037  014536  014552                            RT116                   ;ADDRESS OF NEXT TEST                   *
3038  014540  000144                            100.                    ;ITERATION COUNT                        *
3039  014542  014544                            DAT0                    ;SCOPE ENTRY POINT                      *
3040          000115                            X=X+1
3041                                    ;*****************************************************************************
3042                                    ;DATA TEST 100 CHARACTERS LINE0
3043  014544  004537  006312            DAT0:   JSR     5,DATTST        ;GO RUN DATA TEST
3044  014550  000000                            LINE0                   ;ON LINE0
3045          000001                            Y=Y+1
3046                                    ;*****************************************************************************
3047  014552  000116                    RT116:  116                     ;ROUTINE #116                           *
3048  014554  014570                            RT117                   ;ADDRESS OF NEXT TEST                   *
3049  014556  000144                            100.                    ;ITERATION COUNT                        *
3050  014560  014562                            DAT1                    ;SCOPE ENTRY POINT                      *
3051          000116                            X=X+1
3052                                    ;*****************************************************************************
3053                                    ;DATA TEST 100 CHARACTERS LINE1
3054  014562  004537  006312            DAT1:   JSR     5,DATTST        ;GO RUN DATA TEST
3055  014566  000002                            LINE1                   ;ON LINE1
3056          000002                            Y=Y+1
3057                                    ;*****************************************************************************
3058  014570  000117                    RT117:  117                     ;ROUTINE #117                           *
3059  014572  014606                            RT120                   ;ADDRESS OF NEXT TEST                   *
3060  014574  000144                            100.                    ;ITERATION COUNT                        *
3061  014576  014600                            DAT2                    ;SCOPE ENTRY POINT                      *
3062          000117                            X=X+1
```

```
3063                                    ;*****************************************************************************
3064                                    ;DATA TEST 100 CHARACTERS LINE2
3065  014600  004537  006312            DAT2:   JSR     5,DATTST        ;GO RUN DATA TEST
3066  014604  000004                            LINE2                   ;ON LINE2
3067          000003                            Y=Y+1
3068                                    ;*****************************************************************************
3069  014606  000120                    RT120:  120                     ;ROUTINE #120                              *
3070  014610  014624                            RT121                   ;ADDRESS OF NEXT TEST                      *
3071  014612  000144                            100.                    ;ITERATION COUNT                           *
3072  014614  014616                            DAT3                    ;SCOPE ENTRY POINT                         *
3073          000120                            X=X+1
3074                                    ;*****************************************************************************
3075                                    ;DATA TEST 100 CHARACTERS LINE3
3076  014616  004537  006312            DAT3    JSR     5,DATTST        ;GO RUN DATA TEST
3077  014622  000006                            LINE3                   ;ON LINE3
3078          000004                            Y=Y+1
3079                                    ;*****************************************************************************
3080  014624  000121                    RT121:  121                     ;ROUTINE #121                              *
3081  014626  014642                            RT122                   ;ADDRESS OF NEXT TEST                      *
3082  014630  000144                            100.                    ;ITERATION COUNT                           *
3083  014632  014634                            DAT4                    ;SCOPE ENTRY POINT                         *
3084          000121                            X=X+1
3085                                    ;*****************************************************************************
3086                                    ;DATA TEST 100 CHARACTERS LINE4
3087  014634  004537  006312            DAT4:   JSR     5,DATTST        ;GO RUN DATA TEST
3088  014640  000010                            LINE4                   ;ON LINE4
3089          000005                            Y=Y+1
3090                                    ;*****************************************************************************
3091  014642  000122                    RT122:  122                     ;ROUTINE #122                              *
3092  014644  014660                            RT123                   ;ADDRESS OF NEXT TEST                      *
3093  014646  000144                            100.                    ;ITERATION COUNT                           *
3094  014650  014652                            DAT5                    ;SCOPE ENTRY POINT                         *
3095          000122                            X=X+1
3096                                    ;*****************************************************************************
3097                                    ;DATA TEST 100 CHARACTERS LINE5
3098  014652  004537  006312            DAT5:   JSR     5,DATTST        ;GO RUN DATA TEST
3099  014656  000012                            LINE5                   ;ON LINE5
3100          000006                            Y=Y+1
3101                                    ;*****************************************************************************
3102  014660  000123                    RT123:  123                     ;ROUTINE #123                              *
3103  014662  014676                            RT124                   ;ADDRESS OF NEXT TEST                      *
3104  014664  000144                            100.                    ;ITERATION COUNT                           *
3105  014666  014670                            DAT6                    ;SCOPE ENTRY POINT                         *
3106          000123                            X=X+1
3107                                    ;*****************************************************************************
3108                                    ;DATA TEST 100 CHARACTERS LINE6
3109  014670  004537  006312            DAT6:   JSR     5,DATTST        ;GO RUN DATA TEST
3110  014674  000014                            LINE6                   ;ON LINE6
3111          000007                            Y=Y+1
3112                                    ;*****************************************************************************
3113  014676  000124                    RT124:  124                     ;ROUTINE #124                              *
3114  014700  014714                            RT125                   ;ADDRESS OF NEXT TEST                      *
3115  014702  000144                            100.                    ;ITERATION COUNT                           *
3116  014704  014706                            DAT7                    ;SCOPE ENTRY POINT                         *
3117          000124                            X=X+1
3118                                    ;*****************************************************************************
```

```
3119                                    ;DATA TEST 100 CHARACTERS LINE7
3120  014706  004537  006312            DAT7:   JSR     5,DATTST        ;GO RUN DATA TEST
3121  014712  000016                            LINE7                   ;ON LINE7
3122          000010                            Y=Y+1
3123                                    ;*****************************************************************************
3124  014714  000125                    RT125:  125                     ;ROUTINE #125                           *
3125  014716  014732                            RT126                   ;ADDRESS OF NEXT TEST                   *
3126  014720  000144                            100.                    ;ITERATION COUNT                        *
3127  014722  014724                            DAT10                   ;SCOPE ENTRY POINT                      *
3128          000125                            X=X+1
3129                                    ;*****************************************************************************
3130                                    ;DATA TEST 100 CHARACTERS LINE10
3131  014724  004537  006312            DAT10:  JSR     5,DATTST        ;GO RUN DATA TEST
3132  014730  000020                            LINE10                  ;ON LINE10
3133          000011                            Y=Y+1
3134                                    ;*****************************************************************************
3135  014732  000126                    RT126:  126                     ;ROUTINE #126                           *
3136  014734  014750                            RT127                   ;ADDRESS OF NEXT TEST                   *
3137  014736  000144                            100                     ;ITERATION COUNT                        *
3138  014740  014742                            DAT11                   ;SCOPE ENTRY POINT                      *
3139          000126                            X=X+1
3140                                    ;*****************************************************************************
3141                                    ;DATA TEST 100 CHARACTERS LINE11
3142  014742  004537  006312            DAT11:  JSR     5,DATTST        ;GO RUN DATA TEST
3143  014746  000022                            LINE11                  ;ON LINE11
3144          000012                            Y=Y+1
3145                                    ;*****************************************************************************
3146  014750  000127                    RT127:  127                     ;ROUTINE #127                           *
3147  014752  014766                            RT130                   ;ADDRESS OF NEXT TEST                   *
3148  014754  000144                            100.                    ;ITERATION COUNT                        *
3149  014756  014760                            DAT12                   ;SCOPE ENTRY POINT                      *
3150          000127                            X=X+1
3151                                    ;*****************************************************************************
3152                                    ;DATA TEST 100 CHARACTERS LINE12
3153  014760  004537  006312            DAT12:  JSR     5,DATTST        ;GO RUN DATA TEST
3154  014764  000024                            LINE12                  ;ON LINE12
3155          000013                            Y=Y+1
3156                                    ;*****************************************************************************
3157  014766  000130                    RT130:  130                     ;ROUTINE #130                           *
3158  014770  015004                            RT131                   ;ADDRESS OF NEXT TEST                   *
3159  014772  000144                            100.                    ;ITERATION COUNT                        *
3160  014774  014776                            DAT13                   ;SCOPE ENTRY POINT                      *
3161          000130                            X=X+1
3162                                    ;*****************************************************************************
3163                                    ;DATA TEST 100 CHARACTERS LINE13
3164  014776  004537  006312            DAT13:  JSR     5,DATTST        ;GO RUN DATA TEST
3165  015002  000026                            LINE13                  ;ON LINE13
3166          000014                            Y=Y+1
3167                                    ;*****************************************************************************
3168  015004  000131                    RT131:  131                     ;ROUTINE #131                           *
3169  015006  015022                            RT132                   ;ADDRESS OF NEXT TEST                   *
3170  015010  000144                            100.                    ;ITERATION COUNT                        *
3171  015012  015014                            DAT14                   ;SCOPE ENTRY POINT                      *
3172          000131                            X=X+1
3173                                    ;*****************************************************************************
3174                                    ;DATA TEST 100 CHARACTERS LINE14
```

```
 3175  015014  004537  006312          DAT14:  JSR     5,DATTST        ;GO RUN DATA TEST
 3176  015020  000030                          LINE14                  ;ON LINE14
 3177          000015                          Y=Y+1
 3178                                  ;*****************************************************************************
 3179  015022  000132                  RT132:  132                     ;ROUTINE #132                              *
 3180  015024  015040                          RT133                   ;ADDRESS OF NEXT TEST                      *
 3181  015026  000144                          100.                    ;ITERATION COUNT                           *
 3182  015030  015032                          DAT15                   ;SCOPE ENTRY POINT                         *
 3183          000132                          X=X+1
 3184                                  ;*****************************************************************************
 3185                                  ;DATA TEST 100 CHARACTERS LINE15
 3186  015032  004537  006312          DAT15:  JSR     5,DATTST        ;GO RUN DATA TEST
 3187  015036  000032                          LINE15                  ;ON LINE15
 3188          000016                          Y=Y+1
 3189                                  ;*****************************************************************************
 3190  015040  000133                  RT133:  133                     ;ROUTINE #133                              *
 3191  015042  015056                          RT134                   ;ADDRESS OF NEXT TEST                      *
 3192  015044  000144                          100.                    ;ITERATION COUNT                           *
 3193  015046  015050                          DAT16                   ;SCOPE ENTRY POINT                         *
 3194          000133                          X=X+1
 3195                                  ;*****************************************************************************
 3196                                  ;DATA TEST 100 CHARACTERS LINE16
 3197  015050  004537  006312          DAT16:  JSR     5,DATTST        ;GO RUN DATA TEST
 3198  015054  000034                          LINE16                  ;ON LINE16
 3199          000017                          Y=Y+1
 3200                                  ;*****************************************************************************
 3201  015056  000134                  RT134:  134                     ;ROUTINE #134                              *
 3202  015060  015074                          RT135                   ;ADDRESS OF NEXT TEST                      *
 3203  015062  000144                          100.                    ;ITERATION COUNT                           *
 3204  015064  015066                          DAT17                   ;SCOPE ENTRY POINT                         *
 3205          000134                          X=X+1
 3206                                  ;*****************************************************************************
 3207                                  ;DATA TEST 100 CHARACTERS LINE17
 3208  015066  004537  006312          DAT17:  JSR     5,DATTST        ;GO RUN DATA TEST
 3209  015072  000036                          LINE17                  ;ON LINE17
 3210          000020                          Y=Y+1
 3211                                  ;*************************************************************************
 3212  015074  000135                  RT135:  135                     ;ROUTINE # 135                           *
 3213  015076  015474                          RT136                   ;ADDR OF NEXT ROUTINE.                   *
 3214  015100  000144                          100.                    ;ITERATION COUNT                         *
 3215  015102  015104                          RT135A                  ;SCOPE ENTRY POINT.                      *
 3216          000135                          X=X+1
 3217                                  ;*************************************************************************
 3218
 3219                                  ;TEST THAT DATA (ALL 1'S) CAN BE TRANSMITTED ON LINES SIMULTANEOUSLY
 3220                                  ;THE FOLLOWING TESTS ARE PERFORMED:
 3221                                  ;       THERE ARE 16. DATA ENTRIES
 3222                                  ;       THERE ISN'T A 17TH ENTRY
 3223                                  ;       DATA RECEIVED IS CORRECT
 3224                                  ;       ONE DATA ENTRY PER LINE
 3225
 3226  015104  005037  001600          RT135A: CLR     TUMTAB          ;CLEAR THE
 3227  015110  004537  006224                  JSR     5,BMOVE         ;TUMBLE
 3228  015114  001600                          TUMTAB                  ;TABLE
 3229  015116  001601                          TUMTAB+1                ;(200
 3230  015120  000177                          177                     ;ENTRIES)
```

```
3231  015122  012737  177777  017102          MOV     #-1,OUTBUF      ;LOAD CHAR INTO OUTPUT BUFFER
3232  015130  005000                          CLR     %0              ;SET R0 = LINE 0
3233  015132  012737  000001  001564          MOV     #LBIT0,LINBIT   ;GET LINE BIT
3234  015140  012777  000001  164374          MOV     #BIT0,@CSR      ;SET THE GO BIT
3235  015146  010037  016730          1$:     MOV     %0,LINE         ;GET LINE NUMBER
3236  015152  004537  006246                  JSR     5,XMITD         ;TRANSMIT 1 CHAR.
3237  015156  177777                          -1                      ;ON EACH LINE
3238  015160  005720                          TST     (0)+            ;INCREMENT LINE NUMBER (+2)
3239  015162  006337  001564                  ASL     LINBIT          ;SHIFT LINE BIT TO NEXT LINE
3240  015166  103367                          BCC     1$              ;BRANCH IF ALL LINES NOT DONE
3241  015170  013737  004366  015200          MOV     TIME1,2$        ;PUT TIME TO TRANSMIT 1 CHAR
3242  015176  104400                          DELAY                   ;DELAY 1
3243  015200  000000                  2$      OPEN                    ;CHARACTER TIME
3244  015202  017737  164336  001566          MOV     @BAR,RCVDAT     ;GET & TEST BAR CONTENTS
3245  015210  001410                          BEQ     3$              ;BRANCH IF 0
3246  015212  005037  001570                  CLR     XMTDAT
3247  015216  104011                          ERROR1                  ;ERROR! BAR NOT CLEAR AFTER ALL
3248  015220  005077  164320                  CLR     @BAR            ;LINES FINISHED
3249  015224  005077  164312                  CLR     @CSR
3250  015230  000520                          BR      16$             ;GO TO EXIT
3251  015232  032777  020000  164302  3$      BIT     #BIT13,@CSR     ;TEST THAT OVER RUN DID NOT SET
3252  015240  001404                          BEQ     4$
3253  015242  104001                          ERROR                   ;ERROR! OVER RUN BIT SET
3254  015244  005077  164272                  CLR     @CSR
3255  015250  000510                          BR      16$             ;GO TO EXIT
3256
3257                                  ;TEST THAT THERE ARE 16  VALID DATA ENTRIES
3258  015252  005077  164264          4$:     CLR     @CSR            ;CLEAR THE CSR
3259  015256  012702  000020                  MOV     #16.,%2         ;GET TT SCAN COUNT
3260  015262  012701  001600                  MOV     #TUMTAB,%1      ;GET FIRST TT ADDRESS
3261  015266  005302                  5$:     DEC     %2              ;DECREMENT SCAN COUNTER
3262  015270  100404                          BMI     6$              ;BRANCH IF 16. ENTRIES SCANNED
3263  015272  005721                          TST     (1)+            ;TEST FOR VALID DATA ENTRY
3264  015274  100774                          BMI     5$              ;BRANCH IF FOUND
3265  015276  104001                          ERROR                   ;ERROR! MISING DATA ENTRY
3266  015300  000474                          BR      16$             ;GO TO EXIT
3267  015302  005721                  6$:     TST     (1)+            ;TEST 17TH ENTRY (SHOULD BE = TO 0)
3268  015304  001402                          BEQ     7$              ;BRANCH IF 0
3269  015306  104001                          ERROR                   ;ERROR! EXTRA DATA ENTRY
3270  015310  000470                          BR      16$             ;GO TO EXIT
3271
3272                                  ;TEST THAT THE DATA IS CORRECT IN ALL 16. ENTRIES
3273  015312  012701  001600          7$:     MOV     #TUMTAB,%1      ;GET FIST TT ADDRESS
3274  015316  012702  000020                  MOV     #16.,%2         ;GET SCAN COUNT
3275  015322  005302                  8$:     DEC     %2              ;DECREMENT SCAN COUNT
3276  015324  100421                          BMI     10$             ;BRANCH IF 16. ENTRIES SCANNED
3277  015326  013737  017102  001570          MOV     OUTBUF,XMTDAT   ;GET TRANSMITTED DATA
3278  015334  043737  001572  001570          BIC     CARMSK,XMTDAT   ;CLEAR NON-TRANSMITTED BITS
3279  015342  113737  001600  001566          MOVB    TUMTAB,RCVDAT   ;GET RECEIVED DATA
3280  015350  123737  001570  001566          CMPB    XMTDAT,RCVDAT   ;COMPARE DATA
3281  015356  001402                          BEQ     9$
3282  015360  104011                          ERROR1                  ;ERROR  INCORRECT DATA
3283  015362  000443                          BR      16$             ;GO TO EXIT
3284  015364  005721                  9$:     TST     (1)+            ;INCREMENT TT ADDRESS
3285  015366  000755                          BR      8$              ;TEST NEXT ENTRY
3286
```

```
3287                                    ;CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3288  015370  012701  001600            10$     MOV     #TUMTAB,%1      ;GET FIRST TT ADDRESS
3289  015374  012702  000020                    MOV     #16.,%2         ;GET SCAN COUNT
3290  015400  005302                    11$     DEC     %2              ;DECREMENT SCAN COUNT
3291  015402  100403                            BMI     12$             ;BRANCH IF ALL LINES TESTED
3292  015404  042721  160777                    BIC     #160777,(1)+    ;CLEAR ALL BUT LINE NUMBER IN TT
3293  015410  000773                            BR      11$             ;DO NEXT TT ADDRESS
3294
3295                                    ;TEST THAT THERE IS AN ENTRY FOR EACH OF THE 16  LINES
3296  015412  005037  001570            12$:    CLR     XMTDAT
3297  015416  012701  000020                    MOV     #16.,%1
3298  015422  012702  000020            13$     MOV     #16.,%2
3299  015426  012700  001600                    MOV     #TUMTAB,%0
3300  015432  023720  001570            14$     CMP     XMTDAT,(0)+     ;TEST FOR LINE ENTRY
3301  015436  001406                            BEQ     15$             ;BRANCH IF FOUND
3302  015440  005302                            DEC     %2              ;DECREMENT SCAN COUNT
3303  015442  001373                            BNE     14$             ;LOOK AT NEXT ENTRY
3304  015444  005037  001566                    CLR     RCVDAT
3305  015450  104011                            ERROR1                  ;ERROR! NO ENTRY FOUND FOR THIS LINE
3306  015452  000407                            BR      16$             ;GO TO EXIT
3307  015454  005301            15$.    DEC     %1              ;DECREMENT LINES FOUND COUNT
3308  015456  005701                            TST     %1
3309  015460  001404                            BEQ     16$             ;BRANCH IF ALL LINE TESTED
3310  015462  062737  001000  001570            ADD     #1000,XMTDAT    ;INCREMENT LINE NUMBER
3311  015470  000754                            BR      13$             ;GO DO NEXT LINE
3312  015472  104006            16$:    SCOPE                   ;SCOPE
3313                                    ;*****************************************************************************
3314  015474  000136            RT136:  136                     ;ROUTINE # 136                                  *
3315  015476  016010                    RT137                   ;ADDR OF NEXT ROUTINE                           *
3316  015500  000144                    100.                    ;ITERATION COUNT                                *
3317  015502  015504                    RT136A                  ;SCOPE ENTRY POINT.                             *
3318          000136                    X=X+1
3319                                    ;*****************************************************************************
3320
3321                                    ;TEST THAT THE DM11 CAN TRANSMIT A BREAK ON ALL LINES SIMULTANEOUSLY
3322  015504  013737  004366  015560  RT136A: MOV     @#TIME1,1$      ;GET TIME TO TRANSMIT ONE CHARACTER
3323  015512  005037  001600                    CLR     TUMTAB          ;CLEAR
3324  015516  004537  006224                    JSR     5,BMOVE         ;THE
3325  015522  001600                            TUMTAB                  ;TUMBLE
3326  015524  001601                            TUMTAB+1                ;TABLE
3327  015526  000177                            177
3328  015530  012777  000001  164004            MOV     #BITO,@CSR      ;SET GO
3329  015536  012777  177777  164002            MOV     #-1,@#KCSR      ;SET BREAK BIT FOR ALL LINES
3330  015544  105777  163772                    TSTB    @CSR            ;WAIT FOR THE RECEIVER
3331  015550  100375                            BPL     .-4             ;TO RECEIVE A BREAK
3332  015552  005077  163770                    CLR     @#KCSR          ;CLEAR ALL BREAK BITS
3333  015556  104400                            DELAY                   ;WAIT ONE CHARACTER
3334  015560  000000            1$:     OPEN                    ;TIME
3335  015562  022777  000201  163752            CMP     #201,@CSR       ;TEST THAT ONLY GO AND DONE ARE SET
3336  015570  001410                            BEQ     2$
3337  015572  017737  163744  001566            MOV     @CSR,RCVDAT     ;GET CSR ENTRY
3338  015600  012737  000201  001570            MOV     #201,XMTDAT     ;GET CORRECT RESULT
3339  015606  104011                            ERROR1                  ;ERROR! INCORRECT CSR DATA
3340  015610  000476                            BR      13$             ;EXIT
3341
3342                                    ;TEST THAT THERE IS 16. VALID DATA ENTRIES
```

```
3343  015612  012701  001600          2$      MOV     #TUMTAB,%1      ;GET TUMBLE TABLE BASE ADDRESS
3344  015616  012702  000020                  MOV     #16.,%2         ;GET SCAN COUNT
3345  015622  005721                  3$      TST     (1)+            ;TEST FOR VALID DATA ENTRY
3346  015624  100402                          BMI     4$              ;BRANCH IF VALID DATA ENTRY FOUND
3347  015626  104001                          ERROR                   ;ERROR! MISSING VALID DATA ENTRY
3348  015630  000466                          BR      13$             ;EXIT
3349  015632  005302                  4$      DEC     %2              ;DECREMENT SCAN COUNT
3350  015634  001372                          BNE     3$              ;BRANCH IF 16. ENTRIES NOT SCANNED
3351
3352                                  ;TEST THAT THE BREAK BIT IS SET IN 16. TUMBLE TABLE ENTRIES
3353  015636  012701  001600                  MOV     #TUMTAB,%1
3354  015642  012702  000020                  MOV     #16.,%2
3355  015646  032721  040000          5$      BIT     #BIT14,(1)+     ;BREAK BIT SET?
3356  015652  001002                          BNE     6$              ;BRANCH IF SET
3357  015654  104001                          ERROR                   ;ERROR! MISSING BREAK BIT
3358  015656  000453                          BR      13$             ;EXIT
3359  015660  005302                  6$      DEC     %2              ;DECREMENT SCAN COUNT
3360  015662  001371                          BNE     5$
3361
3362                                  ;TEST THAT THE TUMBLE TABLE DATA BYTE IS ALL 0'S
3363  015664  012701  001600                  MOV     #TUMTAB,%1
3364  015670  012702  000020                  MOV     #16.,%2
3365  015674  105721                  7$:     TSTB    (1)+            ;TEST DATA BYTE
3366  015676  001402                          BEQ     8$              ;BRANCH IF 0'S
3367  015700  104001                          ERROR                   ;ERROR! INCORRECT DATA
3368  015702  000441                          BR      13$             ;EXIT
3369  015704  105721                  8$:     TSTB    (1)+            ;STEP TABLE POINTER TO NEXT DATA BYTE
3370  015706  005302                          DEC     %2
3371  015710  001371                          BNE     7$
3372
3373                                  ;CLEAR ALL BUT LINE NUMBER IN TUMBLE TABLE ENTRY
3374  015712  012701  001600                  MOV     #TUMTAB,%1
3375  015716  012702  000020                  MOV     #16.,%2
3376  015722  042721  160777          9$:     BIC     #160777,(1)+    ;CLEAR ALL BUT LINE NUMBER
3377  015726  005302                          DEC     %2
3378  015730  001374                          BNE     9$
3379
3380                                  ;TEST THAT THERE IS A TUMBLE TABLE ENTRY FOR EACH LINE
3381  015732  005004                          CLR     %4              ;CLEAR LINE NUMBER
3382  015734  012703  000020                  MOV     #16.,%3
3383  015740  012702  000020          10$:    MOV     #16.,%2
3384  015744  012701  001600                  MOV     #TUMTAB,%1
3385  015750  020421                  11$:    CMP     %4,(1)+         ;TEST FOR LINE ENTRY FOR THIS LINE
3386  015752  001410                          BEQ     12$             ;BRANCH IF FOUND
3387  015754  005302                          DEC     %2
3388  015756  001374                          BNE     11$
3389  015760  010437  001570                  MOV     %4,XMTDAT
3390  015764  010437  001566                  MOV     %4,RCVDAT
3391  015770  104011                          ERROR1                  ;ERROR! NO LINE ENTRY FOUND FOR THIS LINE
3392  015772  000405                          BR      13$             ;EXIT
3393  015774  005303                  12$:    DEC     %3              ;ALL LINES BEEN FOUND
3394  015776  001403                          BEQ     13$             ;EXIT IF YES
3395  016000  062704  001000                  ADD     #1000,%4        ;SEARCH FOR
3396  016004  000755                          BR      10$             ;NEXT LINE
3397  016006  104006                  13$:    SCOPE                   ;SCOPE
3398                                  ;******************************************************************************
```

```
 3399  016010  000137                RT137:  137                     ;ROUTINE # 137                  *
 3400  016012  016026                        RT140                   ,ADDR OF NEXT ROUTINE           *
 3401  016014  000002                        2                       ;ITERATION COUNT                *
 3402  016016  016020                        RT137A                  ;SCOPE ENTRY POINT.             *
 3403          000137                        X=X+1
 3404                                ;*************************************************************************
 3405
 3406                                ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
 3407                                ,NEXT LINE.
 3408  016020  004537  005602        RT137A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
 3409  016024  000040                        32.                     ;THIS MUCH BETWEEN LINES
 3410                                ,*************************************************************************
 3411  016026  000140                RT140:  140                     ;ROUTINE # 140                  *
 3412  016030  016044                        RT141                   ;ADDR OF NEXT ROUTINE.          *
 3413  016032  000002                        2                       ;ITERATION COUNT                *
 3414  016034  016036                        RT140A                  ;SCOPE ENTRY POINT.             *
 3415          000140                        X=X+1
 3416                                ;*************************************************************************
 3417
 3418                                ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
 3419                                ;NEXT LINE.
 3420  016036  004537  005602        RT140A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
 3421  016042  000020                        16.                     ;THIS MUCH BETWEEN LINES
 3422                                ;*************************************************************************
 3423  016044  000141                RT141:  141                     ;ROUTINE # 141                  *
 3424  016046  016062                        RT142                   ;ADDR OF NEXT ROUTINE.          *
 3425  016050  000002                        2                       ;ITERATION COUNT                *
 3426  016052  016054                        RT141A                  ;SCOPE ENTRY POINT.             *
 3427          000141                        X=X+1
 3428                                ;*************************************************************************
 3429
 3430                                ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
 3431                                ;NEXT LINE.
 3432  016054  004537  005602        RT141A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
 3433  016060  000010                        8.                      ;THIS MUCH BETWEEN LINES
 3434                                ;*************************************************************************
 3435  016062  000142                RT142:  142                     ;ROUTINE # 142                  *
 3436  016064  016100                        RT143                   ;ADDR OF NEXT ROUTINE.          *
 3437  016066  000002                        2                       ;ITERATION COUNT                *
 3438  016070  016072                        RT142A                  ;SCOPE ENTRY POINT.             *
 3439          000142                        X=X+1
 3440                                ;*************************************************************************
 3441
 3442                                ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
 3443                                ;NEXT LINE.
 3444  016072  004537  005602        RT142A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
 3445  016076  000004                        4                       ;THIS MUCH BETWEEN LINES
 3446                                ;*************************************************************************
 3447  016100  000143                RT143:  143                     ;ROUTINE # 143                  *
 3448  016102  016116                        RT144                   ;ADDR OF NEXT ROUTINE.          *
 3449  016104  000002                        2                       ;ITERATION COUNT                *
 3450  016106  016110                        RT143A                  ;SCOPE ENTRY POINT.             *
 3451          000143                        X=X+1
 3452                                ;*************************************************************************
 3453
 3454                                ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
```

```
3455                                    ;NEXT LINE.
3456  016110  004537  005602            RT143A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
3457  016114  000002                            2                       ;THIS MUCH BETWEEN LINES
3458                                    ;*****************************************************************************
3459  016116  000144                    RT144:  144                     ;ROUTINE # 144                          *
3460  016120  016134                            RT145                   ;ADDR OF NEXT ROUTINE.                  *
3461  016122  000002                            2                       ;ITERATION COUNT                        *
3462  016124  016126                            RT144A                  ;SCOPE ENTRY POINT                      *
3463           000144                           X=X+1
3464                                    ;*****************************************************************************
3465
3466                                    ;TEST TO TRANSMIT ON EACH LINE WITH A DELAY BEFORE STATING THE
3467                                    ;NEXT LINE.
3468  016126  004537  005602            RT144A: JSR     5,@#DLYXMT      ;GO DO TEST. DELAY
3469  016132  000001                            1                       ;THIS MUCH BETWEEN LINES
3470                                    ;*****************************************************************************
3471  016134  000145                    RT145:  145                     ;ROUTINE # 145                          *
3472  016136  016302                            RT146                   ;ADDR OF NEXT ROUTINE.                  *
3473  016140  000144                            100.                    ;ITERATION COUNT                        *
3474  016142  016144                            RT145A                  ;SCOPE ENTRY POINT.                     *
3475           000145                           X=X+1
3476                                    ;*****************************************************************************
3477
3478                                    ;TEST THAT THE DM11 WORKS PROPERLY WHEN THE HALF-DUPLEX BIT (CSR BIT 1)
3479                                    ;IS SET.  THE TEST TRANSMITS DATA ON LINE 0, AND 'BREAKS' ON LINE 1.  ONLY
3480                                    ;THE BREAK SHOULD BE RECEIVED ON LINE 0 IN THE TUMBLE TABLE.
3481  016144  005037  001600            RT145A: CLR     TUMTAB          ;CLEAR THE FIRST TWO
3482  016150  005037  001602                    CLR     TUMTAB+2        ;TUMBLE TABLE ENTRIES
3483  016154  012737  017102  001400            MOV     #OUTBUF,CAT     ;SET UP TO
3484  016162  012737  177777  001440            MOV     #-1,WCT         ;TRANSMIT 1 CHARACTER
3485  016170  012777  000007  163344            MOV     #7,@#CSR        ;SET GO, HALF DUPLEX & MAINT BITS
3486  016176  012777  000001  163340            MOV     #LBIT0,@#BAR    ;TRANSMIT 1 CHAR. ON LINE 0
3487  016204  012777  000002  163334            MOV     #LBIT1,@#KCSR   ;SET BREAK ON LINE 1
3488  016212  105777  163324                    TSTB    @#CSR           ;WAIT FOR THE CHARACTER
3489  016216  100375                            BPL     .-4             ;TO BE RECEIVED
3490  016220  005077  163322                    CLR     @#KCSR          ;CLEAR THE BREAK BIT ON LINE 1
3491
3492                                    ;TEST THAT ONLY THE BREAK WAS RECEIVED
3493  016224  022737  141000  001600            CMP     #141000,TUMTAB  ;TDST FOR BREAK ENTRY (LINE 1)
3494  016232  001410                            BEQ     1$
3495  016234  013737  001600  001566            MOV     TUMTAB,RCVDAT   ;GET ACTUAL ENTRY
3496  016242  012737  141000  001570            MOV     #141000,XMTDAT  ;GET CORRECT ENTRY
3497  016250  104011                            ERROR1                  ;ERROR! INCORRECT BREAK ENTRY
3498  016252  000407                            BR      2$              ;GO TO EXIT
3499  016254  013737  001602  001566  1$:       MOV     TUMTAB+2,RCVDAT ;TEST THAT NEXT ENTRY IS CLEAR
3500  016262  001403                            BEQ     2$              ;EXIT IF CORRECT
3501  016264  005037  001570                    CLR     XMTDAT
3502  016270  104011                            ERROR1                  ;ERROR! SECOND ENTRY WAS NOT CLEAR
3503  016272  005777  163244            2$:     TST     @#CSR           ;WAIT FOR THE TRANSMITTER
3504  016276  100375                            BPL     .-4             ;TO FINISH
3505  016300  104006                            SCOPE                   ;SCOPE
3506                                    ;*****************************************************************************
3507  016302  000146                    RT146:  146                     ;ROUTINE # 146                          *
3508  016304  177777                            RTLAST                  ;ADDR OF NEXT ROUTINE.                  *
3509  016306  000144                            100.                    ;ITERATION COUNT                        *
3510  016310  016312                            RT146A                  ;SCOPE ENTRY POINT.                     *
```

```
        000146                              X=X+1
                                            ;*************************************************************************
                                            
                                            ;TEST THAT THE DM11 RESPONDS CORRECTLY TO A RESET
016312  012737  017102  001400  RT146A: MOV     #OUTBUF,CAT         ;SET UP TO TRANSMIT 10
016320  012737  177770  001440          MOV     #-10,WCT            ;CHARACTERS ON LINE 0
016326  013737  004366  016376          MOV     TIME1,2$            ;GET TIME TO TRANSMIT 2 CHARACTERS
016334  013737  004366  016442          MOV     TIME1,6$
016342  005037  001570                  CLR     XMTDAT
016346  012777  000007  163166          MOV     #7,@CSR             ;SET MAINT., HALF DUPLEX & GO BITS
016354  012777  000001  163162          MOV     #LBIT0,@BAR         ;START TO TRANSMIT ON LINE 0
016362  012777  000002  163156          MOV     #LBIT1,@BKCSR       ;BREAK ON LINE 1
016370  012704  000002                  MOV     #2,%4
016374  104400                  1$:     DELAY                       ;WAIT 2 CHARACTER
016376  000000                  2$      OPEN                        ;TIMES
016400  005304                          DEC     %4
016402  001374                          BNE     1$
016404  104005                          SRESET                      ;RESET
016406  017737  163130  001566          MOV     @CSR,RCVDAT         ;GET CSR CONTENTS
016414  001401                          BEQ     3$                  ;BRANCH IF 0
016416  104011                          ERROR1                      ;ERROR! CSR DID NOT CLEAR
016420  017737  163120  001566  3$.     MOV     @BAR,RCVDAT         ;GET BAR CONTENTS
016426  001402                          BEQ     4$                  ;BRANCH IF 0
016430  104011                          ERROR1                      ;ERROR! BAR IS NOT CLEAR
016432  000427                          BR      5$                  ,EXIT
016434  012704  000010          4$:     MOV     #8.,%4
016440  104400                  5$:     DELAY                       ;WAIT 8 MORE CHARACTER TIMES
016442  000000                  6$:     OPEN
016444  005304                          DEC     %4
016446  001374                          BNE     5$
016450  017737  163066  001566          MOV     @CSR,RCVDAT         ;TEST THAT CSR IS CLEAR
016456  001402                          BEQ     7$
016460  104011                          ERROR1                      ;ERROR! CSR WAS NOT CLEAR
016462  000413                          BR      9$                  ;GO TO EXIT
016464  017737  163054  001566  7$:     MOV     @BAR,RCVDAT         ;TES THAT BAR IS CLEAR
016472  001402                          BEQ     8$
016474  104011                          ERROR1                      ;ERROR! BAR DID NOT CLEAR
016476  000405                          BR      9$
016500  017737  163042  001566  8$:     MOV     @BKCSR,RCVDAT       ;TEST THAT BKCSR IS CLEAR
016506  001401                          BEQ     9$
016510  104011                          ERROR1                      ;ERROR! BKCSR DID NOT CLEAR
016512  104006                  9$      SCOPE                       ;SCOPE
```

```
 3553                                   ;PRG1- TRANSMITTER SCOPE LOOP
 3554  016514                           PRG1:                          ;BEGIN
 3555  016514  104000                           TYPE                   ;TYPE PROGRAM TITLE
 3556  016516  020117                           PRG1M
 3557  016520  004737  016720                   JSR     7,PARAM        ;GO GET USER PARAMETERS
 3558  016524  004737  016772           PRG1R:  JSR     7,LOOP         ;GO LOOP TRANSMITTER
 3559  016530  005777  163010           PRG1B:  TST     @BAR           ;WAIT FOR ALL LINES TO FINISH
 3560  016534  001375                           BNE     PRG1B          ;BRANCH IF NOT DONE
 3561  016536  005077  163003           PRG1C:  CLR     @CSR           ;CLEAR THE CSR
 3562  016542  005037  016570                   CLR     PRG1D          ;CLEAR DELAY TIME
 3563  016546  017737  162430  016570           MOV     @SWR,PRG1D     ;GET DELAY                ,,++D
 3564  016554  042737  000377  016570           BIC     #377,PRG1D                               ,,++D
 3565  016562  000337  016570                   SWAB    PRG1D                                    ,,++D
 3566  016566  104400                           DELAY                  ;DELAY AS SPECIFIED
 3567  016570  000000           PRG1D.  OPEN                           ;BY USER
 3568  016572  000754                           BR      PRG1R          ;LOOP BACK
 3569
 3570
```

```
3571
3572                                        ,PRG2- RECEIVER SCOPE LOOP
3573  016574                                PRG2:                           ,BEGIN
3574  016574  104000                              TYPE                      ,TYPE PROGRAM
3575  016576  020135                              PRG2M                     ;TITLE
3576  016600  004737  016720                      JSR     7,PARAM           ,GO GET USER PARAMETERS
3577  016604  004737  016772                PRG2R: JSR     7,LOOP            ;GO START TRANSMITTER
3578  016610  012777  000001  162724        PRG2A: MOV     #BIT0,@CSR        ,SET GO,CLEAR THE OTHERS
3579  016616  005777  162722                PRG2AA: TST    @BAR              ;HAVE ALL LINES SELECTED FINISHED
3580  016622  001415                              BEQ     PRG2B             ;BRANCH IF FINISHED TRANSMITTING
3581  016624  105777  162712                      TSTB    @CSR              ;WAIT FOR THE RECEIVER TO
3582  016630  100372                              BPL     PRG2AA            ;RECEIVE A CHARACTER
3583  016632  042777  000200  162702              BIC     #BIT7,@CSR        ;CLEAR RECEIVER FLAG
3584  016640  020127  001776                      CMP     %1,#TUMTAB+176    ;IS THE POINTER AT THE END OF THE TT
3585  016644  001002                              BNE      +6               ,BRANCH IF NOT
3586  016646  012701  001576                      MOV     #TUMTAB-2,%1      ,RESET POINTER
3587  016652  005721                              TST     (1)+              ,INCREMENT POINTER
3588  016654  000755                              BR      PRG2A             ,GO BACK & TEST TRANSMITTER FLAG
3589  016656  005077  162660                PRG2B  CLR     @CSR              ,CLEAR THE CSR
3590  016662  005077  162656                       CLR     @BAR              ;CLEAR THE BAR
3591  016666  005037  016714                       CLR     PRG2C             ,CLEAR USER DELAY
3592  016672  017737  162304  016714              MOV     @SWR,PRG2C        ;GET USER DELAY                          ,,++D
3593  016700  042737  000377  016714              BIC     #377,PRG2C                                                 ,,++D
3594  016706  000337  016714                      SWAB    PRG2C                                                      ,,++D
3595  016712  104400                              DELAY                     ;DELAY AS SPECIFIED
3596  016714  000000                        PRG2C OPEN                      ,BY USER
3597  016716  000732                              BR      PRG2R             ;REPEAT LOOP
3598
3599
3600                                        ,SUBROUTINE TO GET USER PARAMETERS (FOR PRG1 & 2)
3601  016720  104000                        PARAM: TYPE                     ;ASK USER WHICH LINE
3602  016722  020154                               LINPAR                   ;TO TEST
3603  016724  004537  004416                      JSR     5,RECD            ;GET LINE AND PUT IT
3604  016730  000000                        LINE:  0                        ;HERE
3605  016732  104000                        PARAMA: TYPE                    ;ASK USER HOW MANY
3606  016734  020177                               HOWMAN                   ;CHARACTERS TO TRANSMIT
3607  016736  004537  004416                      JSR     5,RECD            ;GET CHARS AND PUT IT
3608  016742  000000                        CHARS. 0                        ;HERE
3609  016744  023727  016742  000310              CMP     CHARS,#200.       ;LIMIT RESPONSE TO 200.
3610  016752  101403                              BLOS    PARAMB            ;(CORE LIMITATION)
3611  016754  104000                              TYPE
3612  016756  020017                              M1
3613  016760  000764                              BR      PARAMA            ;RE-REQUEST PARAMETER
3614  016762  104000                        PARAMB: TYPE                    ;TYPE INSTRUCTIONS
3615  016764  020043                              M4
3616  016766  104015                              CNTLU                     ;GO GET VALUE
3617  016770  000207                              RTS     7                 ;EXIT
3618
3619
3620                                        ;SUBROUTINE TO TRANSMIT DATA FROM THE SR
3621  016772  117737  162204  017102        LOOP:  MOVB    @SWR,OUTBUF       ;FILL OUTPUT
3622  017000  004537  006224                      JSR     5,BMOVE           ;BUFFER
3623  017004  017102                              OUTBUF                    ;WITH
3624  017006  017103                              OUTBUF+1                  ;DATA TO BE
3625  017010  000307                              199.                      ;TRANSMITTED
3626  017012  012777  001400  162530              MOV     #CAT,@BASREG      ;INITIALIZE BASE REGISTER
```

```
3627  017020  012737  017102  001400          MOV     #OUTBUF,CAT     ;LOAD CURRENT
3628  017026  004537  006224                  JSR     5,BMOVE         ;ADDRESS TABLE
3629  017032  001400                          CAT                     ;WITH ADDRESS
3630  017034  001402                          CAT+2                   ;OF OUTPUT BUFFER
3631  017036  000040                          32.
3632  017040  013737  016742  001440          MOV     CHARS,WCT       ;LOAD WORD COUNT
3633  017046  005437  001440                  NEG     WCT             ;FORM TWO'S COMPLEMENT
3634  017052  004537  006224                  JSR     5,BMOVE         ;TABLE WITH
3635  017056  001440                          WCT                     ;NUMBER OF
3636  017060  001442                          WCT+2                   ;CHARACTERS TO BE
3637  017062  000040                          32.                     ;TRANSMITTED
3638  017064  013737  016730  001564          MOV     LINE,LINBIT     ;SAVE LINES TO BE TRANSMITTED ON
3639  017072  013777  016730  162444          MOV     LINE,@BAR       ;START TRANSMITTING ON SELECTED LINES
3640  017100  000207                          RTS     7               ;EXIT
3641
3642
3643  017102  000000          OUTBUF: 0                               ;FIRST ADDRESS OF 100
3644          017246                  .=OUTBUF+100.                   ;CHARACTER OUTPUT BUFFER
3645  017246  000000          INBUF:  0                               ;FIRST ADDRESS OF 100.
3646          017412                  .=INBUF+100.                    ;CHARACTER INPUT BUFFER (WHERE RECEIVED
3647                                                                  ;DATA IS STORED)
3648
3649  017412  013746  000006  SUSWRR: MOV     @#6,-(SP)                       ;SAVE VECTORS
3650  017416  013746  000004          MOV     @#4,-(SP)
3651  017422  012737  017442  000004  MOV     #1$,@#4                         ;SET UP FOR TIMEOUT
3652  017430  022777  177777  161544  CMP     #-1,@SWR                        ;REFERENCE HARDWARE SWITCH REGISTER
3653  017436  001402                  BEQ     2$
3654  017440  000407                  BR      3$
3655  017442  022626          1$:     CMP     (SP)+,(SP)+                     ;ADJUST STACK
3656  017444  012737  000176  001202  2$:     MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
3657  017452  012737  000174  001204          MOV     #DISPREG,DISPLAY                ;POINT TO SOFT DISPLAY REG
3658  017460  012637  000004  3$:     MOV     (SP)+,@#4                       ;RESTORE VECTORS
3659  017464  012637  000006          MOV     (SP)+,@#6
3660  017470  000002                  RTI
3661
3662
3663                                  ;ROUTINE TO CHECK FOR  G BEING TYPED
3664
3665  017472  022737  000176  001202  KBDINTT: CMP     #SWREG,SWR
3666  017500  001021                          BNE     1$
3667  017502  023737  000042  000046          CMP     @#42,@#46       ;ACT11?
3668  017510  001415                          BEQ     1$              ;BR IF YES
3669  017512  005037  017604                  CLR     TMP1                    ;CLEAR TEMP AREA
3670  017516  117737  162346  017604          MOVB    @TKDBR,TMP1             ;FETCH THE BUFFER
3671  017524  142737  000200  017604          BICB    #200,TMP1               ;STRIP OFF PARITY
3672  017532  122737  000007  017604          CMPB    #7,TMP1                 ;WAS IT  G
3673  017540  001001                          BNE     1$                      ;NOP
3674  017542  104015                          CNTLU                           ;GO CHANGE IT
3675  017544  000002          1$:     RTI                             ;EXIT
3676
3677
3678                                  ;ROUTINE TO CHANGE CONTENTS OF SWREG(LOC 176)
3679
3680  017546  022737  000176  001202  CNTLUU: CMP     #SWREG,SWR
3681  017554  001023                          BNE     FJX
3682  017556  104000                          TYPE
```

```
 3683  017560  017641                        $SWREG
 3684  017562  004537  006154                JSR     R5,OACNV              ,CONVERT TO ASCII
 3685  017566  000176                        SWREG
 3686  017570  017647                        $VALUE
 3687  017572  000006                        6
 3688  017574  104000                        TYFE
 3689  017576  017647                        $VALUE
 3690  017600  004537  004416                JSR     5,RECD                ,GET TMP1 AND PUT IT
 3691  017604  000000              TMP1.     0                             ,HERE
 3692  017606  022737  000007  004574        CMP     #7,CNT
 3693  017614  001403                        BEQ     FJX
 3694  017616  013777  017604  161356        MOV     TMP1,@SWR             ,CHANGE CONTENTS OF SWREG
 3695  017624  000002              FJX:      RTI
 3696
```

```
                                         ,MESSAGES
017626  053045  041505  020124  WHERE    .ASCII   '%VECT ADR?@'
017634  042101  037522     100
017641     045  053523  036522  $SWREG:  .ASCII   '%SWR=@'
017646     100
017647     040  020040  020040  $VALUE:  .ASCII   '          NEW=@'
017654  020040  020040  047040
017662  053505  040075
017666  044445  041516  051117  $CTLU    .ASCII   '%INCORRECT INPUT, TRY AGAIN!'
017674  042522  052103  044440
017702  060116  052125  020054
017710  051124  020131  043501
017716  044501  020516
017722  036445     100                    .ASCII   '%=@'
017725     045  047125  052111  WHICH:   .ASCII   '%UNIT#(8)?@'
017732  024043  024470  040077
017740  041445  040510  020122  LEVEL    .ASCII   '%CHAR LNGTH@'
017746  047114  052107  040110
017754  042445  051122  051440  ERDAT    .ASCII   '%ERR S/B  '
017762  041057  020072
017766  020040  020040  020040  AASB:    .ASCII   '       WAS  '
017774  063440  051501  020072
020002  020040  020040  020040  AWAS     .ASCII   '      @'
020010     100
020011     045  051120  021507  M0:      .ASCII   '%PRG#@'
020016     100
020017     045  040077          M1:      .ASCII   '%?@'
020022  042445  042116  040040  M2:      .ASCII   '%END @'
020030  051445  036522  037460  M3:      .ASCII   '%SR=0? GO @'
020036  043440  027117     100
020043     045  042114  041440  M4:      .ASCII   '%LD CHAR IN SR0-7,DLY IN SR8-15@'
020050  040510  020122  047111
020056  061440  030122  033455
020064  042073  064514  044440
020072  020116  051123  026470
020100  032461     100
020103     045  047514  044507  PRG0M:   .ASCII   '%LOGIC TSTS@'
020110  020103  051524  051524
020116     100
020117     045  046530  052111  PRG1M:   .ASCII   '%XMITTER LOOP@'
020124  042524  020122  047514
020132  060117     100
020135     045  046530  052111  PRG2M:   .ASCII   '%XMIT/REC LOOP@'
020142  061057  041506  046040
020150  047517  040120
020154  062046  060131  046040  LINPAR:  .ASCII   '%TYP LINES TO TST @'
020162  047111  061506  052040
020170  020117  061524  020124
020176     100
020177     045  047443  020106  HOWMAN:  .ASCII   '%#OF CHARS?@'
020204  044103  051101  037523
020212     100
020213     045  020122          EM0:     .ASCII   '%R '
020216  020040  050040  036503  ATNUMB:  .ASCII   '   PC='
020224  020040  020040  020040  APC:     .ASCII   '      @'
020232     100
```

```
 3753  020233      015      012          MSG1    .BYTE   15,12
 3754  020235      124  042510  050440           ASCII  'THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS BACK 1234567890'
 3755  020242  044525  045503  041040
 3756  020250  047522  047127  043040
 3757  020256  054117  045040  046525
 3758  020264  042520  020104  053117
 3759  020272  051105  052040  042510
 3760  020300  046040  055101  020131
 3761  020306  047504  051507  041040
 3762  020314  041501  020113  031061
 3763  020322  032063  033065  034067
 3764  020330  030071
 3765          000001                           END
```

```
A       = 000000            526#  2341#  2855#  3033#
AASB      017766            802   3717#
APC       020224            771   3751#
APCADD    002102            736#   761#  764#   770
AREA      013526           2689   2700   2719#  2720   2730   2740#
ATNUMB    020216            775   3750#
AWAS      020002            806   3719#
A12K      013546           2749#
A16K      013552           2751#
A20K      013556           2753#
A24K      013562           2755#
A28K      013566           2757#
A8K       013536           2744#
BAR       001544            688#   882#  1033#  1131#  1245   1435   1523#  2008   2036#  2038   2042#  2043#  2044
                           2081#  2082#  2083   2086   2089#  2543#  2547   2551#  2589#  2592#  2653#  2692#  2723#
                           2787   2789   2801   2803   2833#  3244   3248#  3486#  3521#  3532   3545   3559   3579
                           3590#  3639#
BASREG    001550            690#  1009#  1520#  1951#  1952   1960#  1961   1982   1986#  1987   1993#  1994#  2009#
                           2585#  2648#  3626#
BAT       001500            684#   685
BELL    = 000007            521#   786
BIT0    = 000001            469#  1031   1132   1305   1417   1544   1698   1699   1703   2832   3234   3328   3578
BIT1    = 000002            468#  1717   1718   1722
BIT10   = 002000            459#
BIT11   = 004000            458#   890
BIT12   = 010000            457#  1261   1269   1812   1813   1817   2130   2558
BIT13   = 020000            456#   767   1332   1362   1547   1552   1568   1831   1832   1836   2131   3251
BIT14   = 040000            455#   887   1393   1547   1552   1881   1889   1915   1956   1964   1991   2553   2593
                           2665  3355
BIT15   = 100000            437   454#  1269   1308   1336   1366   1547   1556   2557
BIT2    = 000004            467#  1031   1132   1736   1737   1741   2828
BIT3    = 000010            466#
BIT4    = 000020            465#  1755   1756   1760
BIT5    = 000040            464#  1774   1775   1779
BIT6    = 000100            463#  1031   1132   1318   1326   1793   1794   1798
BIT7    = 000200            462#  1326   1355   1384   1387   1559   2777   2836   3583
BIT8    = 000400            461#   782
BIT9    = 001000            460#   865    898
BKCSR     001546            689#   884#  1381#  1388#  1876#  1877   1885#  1886   1907   1910#  1911   1917#  1918#
                          1933#  1936   2105#  2106#  2107   2110   2113#  3329#  3332#  3487#  3490#  3522#  3549
BMOVE     006224           1411   1464   1505#  1539   3227   3324   3622   3628   3634
BRKTST    005456           1380#  2865   2876   2887   2898   2909   2920   2931   2942   2953   2964   2975   2986
                           2997   3008   3019   3030
BRK0      014204           2861   2865#
BRK1      014222           2872   2876#
BRK10     014364           2949   2953#
BRK11     014402           2960   2964#
BRK12     014420           2971   2975#
BRK13     014436           2982   2986#
BRK14     014454           2993   2997#
BRK15     014472           3004   3008#
BRK16     014510           3015   3019#
BRK17     014526           3026   3030#
BRK2      014240           2883   2887#
BRK3      014256           2894   2898#
BRK4      014274           2905   2909#
```

```
BRK5    014312          2916    2920#
BRK6    014330          2927    2931#
BRK7    014346          2938    2942#
CARMSK  001572           699#   1114#  1334   1364   1603   2701   2731   3278
CAT     001400           680#    681   1009   1030#  1128#  1256   1520   1521#  2009   2534   2648   2649#  2659
                        2663    2689#  2720#  3483#  3515#  3626   3627#  3629   3630
CC      = 177776         432#
CHALT   = 104003         509#
CHARS   016742          3608#   3609   3632
CLKINT  001552           691#   1061#  1129#  1317#
CLKLVL  001554           692#   1130#
CNT     004574          1175#   1192#  1211#  3692
CNTLU   = 104015         519#    860   3616   3674
CNTLUU  017546           729    3680#
CNVOAT  002332           753     765    800#
COUNT   002076           734#   1034#  1035#  1134#  1141#  1409#  1421   1531#  1535#
CSR     001542           687#    883#  1031#  1081   1088   1132#  1137#  1144#  1248   1261#  1269#  1270   1305#
                        1306    1308#  1310   1318#  1326#  1327   1351   1353   1355#  1380#  1382   1384#  1385
                        1387#  1417#  1439   1442   1544#  1547   1549   1552   1556#  1557   1559#  1682   1698#
                        1699    1703#  1704   1717#  1718   1722#  1723   1736#  1737   1741#  1742   1755#  1756
                        1760#  1761   1774#  1775   1779#  1780   1793#  1794   1798#  1799   1812#  1813   1817#
                        1818    1831#  1832   1836#  1837   1851#  1854   1858#  1859#  1860   2057#  2058#  2059
                        2062    2065#  2069   2093   2117   2130#  2131#  2149#  2157#  2158#  2542#  2553   2557#
                        2558#  2567#  2593   2617   2619#  2620   2652#  2654   2656#  2657   2690#  2693   2695
                        2697#  2721#  2724   2726   2728#  2770#  2775   2777#  2781   2797   2828#  2832#  2834
                        2836#  2847   2849   3234#  3249#  3251   3254#  3258#  3328#  3330   3335   3337   3485#
                        3488   3503   3520#  3529   3541   3561#  3578#  3581   3583#  3589#
CURTST  002002           704#    867    873    917#
DATCHK= 104002           508#   1605
DATTST  006312          1530#   3043   3054   3065   3076   3087   3098   3109   3120   3131   3142   3153   3164
                        3175    3186   3197   3208
DAT0    014544          3039    3043#
DAT1    014562          3050    3054#
DAT1AA  006316          1531#
DAT10   014724          3127    3131#
DAT11   014742          3138    3142#
DAT12   014760          3149    3153#
DAT13   014776          3160    3164#
DAT14   015014          3171    3175#
DAT15   015032          3182    3186#
DAT16   015050          3193    3197#
DAT17   015066          3204    3208#
DAT2    014600          3061    3065#
DAT3    014616          3072    3076#
DAT4    014634          3083    3087#
DAT5    014652          3094    3098#
DAT6    014670          3105    3109#
DAT7    014706          3116    3120#
DELAY   = 104400         520#   1422   1431   2545   2573   2590   2598   2630   2795   3242   3333   3524   3537
                        3566    3595
DISPLA  001204           678#   3657#
DISPRE  000174           666#   3657
DLY     006000           542    1448#
DLYXMT  005602          1409#   3408   3420   3432   3444   3456   3468
DMPAR   003464           831    1018#  1056
DMPARB  003746          1066#   1076
```

```
DMPARC  004072          1095#  1108
DTCHK   002126           718    751#
EMALT = 104004           510#
EMO     020213           778   3749#
EMTINT  003100           540    922#
EMTTAB  002032           716#   928
ERDAT   017754           781   3715#
ERR     002146           717    760#
ERRA    002204           762    766#
ERRB    002246           760*   763*   779#
ERRC    002254           768    782#
ERRD    002306           783    789#
ERREX   002326           792    794#
ERRHLT  002324           790    793#
ERROR = 104001           507#  1037   1139   1220   1250   1265   1272   1314   1322   1329   1391   1395   1554
                        1562  1623   1627   1687   1701   1706   1720   1725   1739   1744   1758   1763   1777
                        1782  1796   1801   1815   1820   1834   1839   2134   2152   2161   2555   2595   2799
                        2853  3253   3265   3269   3347   3357   3367
ERROR1= 104011           515#   754   1247   1254   1259   1341   1347   1359   1370   1402   1437   1443   1580
                        1856  1862   1880   1888   1914   1938   1955   1963   1990   2022   2040   2046   2063
                        2070  2087   2094   2111   2118   2550   2564   2624   2664   2704   2734   2783   2790
                        2805  2841   3247   3282   3305   3339   3391   3497   3502   3531   3534   3543   3547
                        3551
ERRVEC= 000004           434#   820*   827*   862*  1681*  1683*  2586*  2600*  2646*  2647*  2673*  2674*  2714*
                        2737*
ERR1    002164           725    763#
ESCOPE  002700           722    880#
FJX     017624          3681   3693   3695#
FORMD   003046           864    912#
FORMDA  003072           917#
GETRDY  002600           855    859    861#   877    899    906   1671
GTBIN   006104          1474#  1533
GTLINB  007044          1235   1519   1653#
GTRDYA  002616           864#   875
GTRDYC  002640           866    869#
GTRDYD  002662           872    874#
GTRDYX  002606           862#   901
HOLDBN  020177          3606   3746#
ICTR    002010           707#   896*   915*
INBIN   006064          1463#
INBUF   017246          1543   1595   3645#  3646
INCRTN  002120           746#   876
INIT    003434           726   1009#
INITIA= 104012           516#   833    849    863    885
KBDIN = 104014           518#   766    794    881
KBDINT  017472           728   3665#
KSTART  002000           703#   861   1666*  1669*
LBIT0 = 000001           453#  1033   1131   1415   2540   2589   2653   2692   2723   2833   2866   3233   3486
                        3521
LBIT1 = 000002           452#  2877   3487   3522
LBIT10= 000400           445#  2954
LBIT11= 001000           444#  2965
LBIT12= 002000           443#  2976
LBIT13= 004000           442#  2987
LBIT14= 010000           441#  2998
LBIT15= 020000           440#  3009
```

```
LBIT16= 040000          439#  3020
LBIT17= 100000          438#  3031
LBIT2 = 000004          451#  2888
LBIT3 = 000010          450#  2899
LBIT4 = 000020          449#  2910
LBIT5 = 000040          448#  2921
LBIT6 = 000100          447#  2932
LBIT7 = 000200          446#  2943
LENGTH  004112         1096   1100#  1101   1103#  1104   1109   1111#  1112#  1113
LENOKA  004130         1102   1104#
LENOKB  004140         1106#  1110
LENOKC  004146         1105   1109#
LENOKD  004206         1114   1120#
LEVEL   017740         1098   3713#
LINBIT  001564          696#  1415#  1427#  1523   1654#  1657#  2622   2623   2627#  2787   2808#  3233#  3239#
                       3638#
LINE    016730         1234#  1251   1302#  1335   1365   1416#  1426#  1518   1530#  1576   1578   1588   1655
                       2614#  2626#  2769#  2772   2807#  3235#  3604#  3638   3639
LINE0 = 000000          481#  2173   2351   3044
LINE1 = 000002          482#  2184   2362   3055
LINE10= 000020          489#  2261   2439   3132
LINE11= 000022          490#  2272   2450   3143
LINE12= 000024          491#  2283   2461   3154
LINE13= 000026          492#  2294   2472   3165
LINE14= 000030          493#  2305   2483   3176
LINE15= 000032          494#  2316   2494   3187
LINE16= 000034          495#  2327   2505   3198
LINE17= 000036          496#  2338   2516   3209
LINE2 = 000004          483#  2195   2373   3066
LINE3 = 000006          484#  2206   2384   3077
LINE4 = 000010          485#  2217   2395   3088
LINE5 = 000012          486#  2228   2406   3099
LINE6 = 000014          487#  2239   2417   3110
LINE7 = 000016          488#  2250   2428   3121
LINPAR  020154         3602   3742#
LOOP    016772         3558   3577   3621#
LTST0   011454         2168   2172#
LTST1   011472         2179   2183#
LTST10  011634         2256   2260#
LTST11  011652         2267   2271#
LTST12  011670         2278   2282#
LTST13  011706         2289   2293#
LTST14  011724         2300   2304#
LTST15  011742         2311   2315#
LTST16  011760         2322   2326#
LTST17  011776         2333   2337#
LTST2   011510         2190   2194#
LTST3   011526         2201   2205#
LTST4   011544         2212   2216#
LTST5   011562         2223   2227#
LTST6   011600         2234   2238#
LTST7   011616         2245   2249#
MACHER  000004          530#
MANUAL= 100000          437#
MSG1    020233         1412   3753#
MO      020011          837   3721#
```

```
M1        020017            747    1055    1075    1107    1207    3612    3723#
M2        020022            904    3724#
M3        020030            857    3725#
M4        020043           3615    3727#
NOP    =  000240            435#
NXTST     002006            706#    861*    874     900     912     914*
OACNV     006154            769     773     800     804    1486#   3684
OACNVA    006166           1491#   1500
OPEN   =  000000            436#    680     682     684     686     691     693     695     696     697     698     699     701
                            703     704     705     706     707     708     709     734     735     736     737     742     779
                            942     943     957     958    1004    1152    1153    1459    1460    1469    1470    1471    1644
                           1645    1646    1647    1648    1649    1650    2546    2574    2591    2599    2796    3243    3334
                           3525    3538    3567    3596
OUTBUF    017102           1030    1128    1255    1298*   1413    1521    1532    1594    1600    3231*   3277    3483    3515
                           3621*   3623    3624    3627    3643#   3644
OVRLAY    004372           1042    1044    1156#
PARAM     016720           3557    3576    3601#
PARAMA    016732           3605#   3613
PARAMB    016762           3610    3614#
PAR0      007026           1633*   1644#
PAR1      007030           1634*   1645#
PAR2      007032           1635*   1646#
PAR3      007034           1636*   1647#
PAR4      007036           1637*   1648#
PAR5      007040           1638*   1649#
PAR6      007042           1639*   1650#
PASS      002116            742#    894     902*
PCADD     002100            735#    761     764     924*
PFAIL     004576            842     846    1214#
POPSP  =  005726            470#   1444    1629
POPSP2=   022626            471#   1149    2736
PRGLIM    002014            709#    840
PRGNUM    002454            835     839#    840*    841*    844     850
PRGTAB    002016            710#    845
PRG0      007076            710    1664#
PRG0M     020103           1665    3733#
PRG0R     007120            713    1669#
PRG1      016614            711    3554#
PRG1B     016630           3559#   3560
PRG1C     016636           3561#
PRG1D     016670           3562*   3563*   3564*   3565*   3567#
PRG1M     020117           3556    3736#
PRG1R     016624            714    3558#   3568
PRG2      016674            712    3573#
PRG2A     016610           3578#   3588
PRG2AA    016616           3579#   3582
PRG2B     016666           3580    3589#
PRG2C     016714           3591*   3592*   3593*   3594*   3596#
PRG2M     020135           3575    3739#
PRG2R     016604            715    3577#   3597
PRTY0  =  000000            479#
PRTY1  =  000040            478#
PRTY2  =  000100            477#
PRTY3  =  000140            476#
PRTY4  =  000200            475#   2150
PRTY5  =  000240            474#    692     694
```

```
PRTY6 = 000300            473#
PRTY7 = 000340            472#    541     543     843     847    1010    1130    1138    1264    1268    1301    1321    1324
                         2147    2561    2568    2647
PRVCNT  002104            737#   1907#   1918    1982#   1994
PSW   = 177776            433#   1010#   1032#   1133#   1138#   1262#   1264#   1268#   1301#   1319#   1321#   1324#   2132#
                         2147#   2150#   2155#   2559#   2561#   2568#
PT0     006100           1470#   1474    1480#
PT1     006102           1471#   1474#   1475#   1478#   1479#   1480    1481
PWRUP   004606           1214    1218#
RCVDAT  001566            697#    751     805    1245#   1252#   1256#   1257    1331#   1332#   1337    1344#   1356#   1361#
                         1362#   1367    1399#   1400#   1435#   1442#   1579#   1598#   1604#   1610#   1617#   1854#   1860#
                         1877#   1878    1886#   1911#   1912    1936#   1952#   1953    1961#   1987#   1988    2021#   2038#
                         2044#   2062#   2069#   2086#   2093#   2110#   2117#   2547#   2563#   2622#   2663#   2698#   2699#
                         2702    2729#   2732    2771#   2778#   2779    2803#   2830#   2837#   3244#   3279#   3280    3304#
                         3337#   3390#   3495#   3499#   3529#   3532#   3541#   3545#   3549#
RCVTST  005034           1298#   2350    2361    2372    2383    2394    2405    2416    2427    2438    2449    2460    2471
                         2482    2493    2504    2515
RCV0    012014           2346    2350#
RCV1    012032           2357    2361#
RCV10   012174           2434    2438#
RCV11   012212           2445    2449#
RCV12   012230           2456    2460#
RCV13   012246           2467    2471#
RCV14   012264           2478    2482#
RCV15   012302           2489    2493#
RCV16   012320           2500    2504#
RCV17   012336           2511    2515#
RCV2    012050           2368    2372#
RCV3    012066           2379    2383#
RCV4    012104           2390    2394#
RCV5    012122           2401    2405#
RCV6    012140           2412    2416#
RCV7    012156           2423    2427#
RECD    004416            838    1047    1070    1099    1173#   3603    3607    3690
RIND    006076           1463#   1465    1466    1469#   1476#
RNGEN   003250            968#   1908    1983
RORPAR  006764           1571    1597    1619    1633#   1641
RP1     003316            968     972#    979     981#
RP2     003320            971     975     978#    982#
RSTART  002024            713#    851
RSTAT1  002426            670     832#
RSTAT2  002614            672     846#   1221
RSTPC   003226            946#    955     957#
RSTPSW  003230            947#    954     958#
RSTREG= 104010            514#   1501    1512
RSTRG   003172            724     946#
RTLAST= 177777            522#   3508
RTNMO   002004            705#    774     871     913#   1667#   1670#
RT0     007136           1666    1669    1673#
RT0A    007146           1676    1681#
RT1     007200           1674    1690#
RT1A    007210           1693    1698#
RT10    007646           1805    1823#
RT10A   007656           1826    1831#
RT100   014246           2881    2891#
RT101   014264           2892    2902#
```

RT102	014302	2903	2913#	
RT103	014320	2914	2924#	
RT104	014336	2925	2935#	
RT105	014354	2936	2946#	
RT106	014372	2947	2957#	
RT107	014410	2958	2968#	
RT11	007720	1824	1842#	
RT11A	007730	1845	1851#	1857
RT110	014426	2969	2979#	
RT111	014444	2980	2990#	
RT112	014462	2991	3001#	
RT113	014500	3002	3012#	
RT114	014516	3013	3023#	
RT115	014534	3024	3036#	
RT116	014552	3037	3047#	
RT117	014570	3048	3058#	
RT12	010010	1843	1866#	
RT12A	010020	1869	1875#	
RT120	014606	3059	3069#	
RT121	014624	3070	3080#	
RT122	014642	3081	3091#	
RT123	014660	3092	3102#	
RT124	014676	3103	3113#	
RT125	014714	3114	3124#	
RT126	014732	3125	3135#	
RT127	014750	3136	3146#	
RT13	010144	1867	1898#	
RT13A	010154	1901	1906#	
RT130	014766	3147	3157#	
RT131	015004	3158	3168#	
RT132	015022	3169	3179#	
RT133	015040	3180	3190#	
RT134	015056	3191	3201#	
RT135	015074	3202	3212#	
RT135A	015104	3215	3226#	
RT136	015474	3213	3314#	
RT136A	015504	3317	3322#	
RT137	016010	3315	3399#	
RT137A	016020	3402	3408#	
RT14	010256	1899	1924#	
RT14A	010266	1927	1933#	
RT140	016026	3400	3411#	
RT140A	016036	3414	3420#	
RT141	016044	3412	3423#	
RT141A	016054	3426	3432#	
RT142	016062	3424	3435#	
RT142A	016072	3438	3444#	
RT143	016100	3436	3447#	
RT143A	016110	3450	3456#	
RT144	016116	3448	3459#	
RT144A	016126	3462	3468#	
RT145	016134	3460	3471#	
RT145A	016144	3474	3481#	
RT146	016302	3472	3507#	
RT146A	016312	3510	3515#	
RT15	010316	1925	1941#	

RT15A	010326	1944	1950#
RT16	010452	1942	1973#
RT16A	010462	1976	1981#
RT17	010570	1974	2000#
RT17A	010600	2003	2008#
RT2	007252	1691	1709#
RT2A	007262	1712	1717#
RT20	010662	2001	2027#
RT20A	010672	2030	2035#
RT21	010750	2028	2049#
RT21A	010760	2052	2057#
RT22	011056	2050	2073#
RT22A	011066	2076	2081#
RT23	011164	2074	2097#
RT23A	011174	2100	2105#
RT24	011272	2098	2121#
RT24A	011302	2124	2129#
RT25	011336	2122	2139#
RT25A	011346	2142	2147#
RT26	011444	2140	2165#
RT27	011462	2166	2176#
RT3	007324	1710	1728#
RT3A	007334	1731	1736#
RT30	011500	2177	2187#
RT31	011516	2188	2198#
RT32	011534	2199	2209#
RT33	011552	2210	2220#
RT34	011570	2221	2231#
RT35	011606	2232	2242#
RT36	011624	2243	2253#
RT37	011642	2254	2264#
RT4	007376	1729	1747#
RT4A	007406	1750	1755#
RT40	011660	2265	2275#
RT41	011676	2276	2286#
RT42	011714	2287	2297#
RT43	011732	2298	2308#
RT44	011750	2309	2319#
RT45	011766	2320	2330#
RT46	012004	2331	2343#
RT47	012022	2344	2354#
RT5	007450	1748	1766#
RT5A	007460	1769	1774#
RT50	012040	2355	2365#
RT51	012056	2366	2376#
RT52	012074	2377	2387#
RT53	012112	2388	2398#
RT54	012130	2399	2409#
RT55	012146	2410	2420#
RT56	012164	2421	2431#
RT57	012202	2432	2442#
RT6	007522	1767	1785#
RT6A	007532	1788	1793#
RT60	012220	2443	2453#
RT61	012236	2454	2464#
RT62	012254	2465	2475#

```
RT63    012272          2476    2486#
RT64    012310          2487    2497#
RT65    012326           822#   2498    2508#
RT66    012346          2509    2520#
RT66A   012362          2523    2532#
RT67    012606          2521    2577#
RT67A   012616          2580    2585#
RT7     007574          1786    1804#
RT7A    007604          1807    1812#
RT70    012716           822    2578    2603#
RT70A   012726          2606    2614#
RT71    013032          2604    2634#
RT71A   013042          2637    2643#
RT72    013226          2635    2677#
RT72A   013236          2680    2687#
RT73    013572          2678    2760#
RT73A   013602          2763    2769#
RT74    014006          2761    2812#
RT74A   014016          2815    2822#
RT75    014174          2813    2858#
RT76    014212          2859    2869#
RT77    014230          2870    2880#
SAVREG= 104007           513#   1486    1505
SAVRG   003132           723     932#
SCOPE = 104006           512#   1274    1371    1403    1445    1630    1684    1707    1726    1745    1764    1783    1802
                        1821    1840    1864    1896    1922    1939    1971    1998    2019    2047    2071    2095    2119
                        2137    2163    2575    2601    2632    2675    2738    2810    2854    3312    3397    3505    3552
SCOPEA  002736           889#    897
SCOPEB  002742           888     890#
SCOPED  002776           868     891     895     898#
SCOPTR  002012           708#    886     916*
SPBOT   001200           676#    818     825     832     848    1011*   1012    1219
SRESET= 104005           511#   1853    1935    2037    3528
SRSET   002546           852#   1668
SRSETT  003232           721     961#    963*
START   002360           669     818#
SUSWR = 104013           517#    819
SUSWRR  017412           727    3649#
SVRPC   003166           932*    940     942#
SVRPSW  003170           933*    939     943#
SWR     001202           677#    767     782     791     852     865     869     887     890     898    1881    1889    1915
                        1956    1964    1991    2665    3563    3592    3621    3652    3656*   3665    3680    3694*
SWREG   000176           666#    852    3656    3665    3680    3685
TIMEA   004272          1134#   1143
TIMEB   004326          1136    1141#
TIMEC   004336          1129    1144#
TIMER   004216          1126#   1140    2532
TIME1   004366          1126*   1135*   1145    1152#   1420    1430    2572    2597    2629    3241    3322    3517    3518
TIME1A  004370          1145*   1146*   1148*   1153#   2544    2588    2794
TKCSR   002066           730#   1176
TKDBR   002070           731#   1178    3670
TMP1    017604          3669*   3670*   3671*   3672    3691#   3694
TPCSR   002072           732#    784     787     996    1195    1198    1201
TPDBR   002074           733#    786*    995*   1180*   1197*   1200*
TTDAT   001562           695#   1574*   1575*   1576    1579
TUMTAB  001600           701#    702    1299*   1300*   1331    1343*   1344    1356    1361    1389    1393    1397    1400
```

```
                    1537    1538#   1540    1541    1583    1585    2699    2729    2822    2827    2851    3226#   3228
                    3229    3260    3273    3279    3288    3299    3323#   3325    3326    3343    3353    3363    3374
                    3384    3481#   3482#   3493    3495    3499    3584    3586
TYP         003322   716     984#
TYPA        003332   987#    994     1003
TYPC        003350   989     991#
TYPD        003366   993     995#    1000    1002
TYPDAT      003432   987#    988     991     995     999#    1001#   1004#
TYPE    =   104000   506#    746     777     780     836     856     903     1045    1054    1068    1074    1097    1106
                    1206    1208    1664    3555    3574    3601    3605    3611    3614    3682    3688
TYPF        003404   992     999#
TYPG        003416   1001#
UNIT        003766   1067    1071#   1072    1077#   1078#   1079#   1083
UNTOKA      004006   1073    1077#
UNTOKB      004032   1082#   1085
UNTOKC      004062   1090#   1093
VAC         001540   686#    700
VECOK       003666   1043    1050    1052#
VECOKA      003676   1054#   1058    1060
VECOKB      003704   1053    1057#
VECTOR      003650   1040#   1048#   1049    1051#   1052    1057    1059    1061    1062#   1063
WCT         001440   682#    683     1029#   1127#   1252    1522#   1589    2651#   2691#   2722#   2831#   3484#   3516#
                    3632#   3633#   3635    3636
WHERE       017626   1046    3698#
WHICH       017725   1069    3711#
X       =   000146    525#   1672    1677#   1689    1694#   1708    1713#   1727    1732#   1746    1751#   1765    1770#
                     1784   1789#   1803    1808#   1822    1827#   1841    1846#   1865    1870#   1897    1902#   1923
                     1928#  1940    1945#   1972    1977#   1999    2004#   2026    2031#   2048    2053#   2072    2077#
                     2096   2101#   2120    2125#   2138    2143#   2164    2169#   2175    2180#   2186    2191#   2197
                     2202#  2208    2213#   2219    2224#   2230    2235#   2241    2246#   2252    2257#   2263    2268#
                     2274   2279#   2285    2290#   2296    2301#   2307    2312#   2318    2323#   2329    2334#   2342
                     2347#  2353    2358#   2364    2369#   2375    2380#   2386    2391#   2397    2402#   2408    2413#
                     2419   2424#   2430    2435#   2441    2446#   2452    2457#   2463    2468#   2474    2479#   2485
                     2490#  2496    2501#   2507    2512#   2518    2519    2524#   2576    2581#   2602    2607#   2633
                     2638#  2676    2681#   2759    2764#   2811    2816#   2857    2862#   2868    2873#   2879    2884#
                     2890   2895#   2901    2906#   2912    2917#   2923    2928#   2934    2939#   2945    2950#   2956
                     2961#  2967    2972#   2978    2983#   2989    2994#   3000    3005#   3011    3016#   3022    3027#
                     3035   3040#   3046    3051#   3057    3062#   3068    3073#   3079    3084#   3090    3095#   3101
                     3106#  3112    3117#   3123    3128#   3134    3139#   3145    3150#   3156    3161#   3167    3172#
                     3178   3183#   3189    3194#   3200    3205#   3211    3216#   3313    3318#   3398    3403#   3410
                     3415#  3422    3427#   3434    3439#   3446    3451#   3458    3463#   3470    3475#   3506    3511#
XMITD       006246   1237    1303    1349    1418    1517#   1545    2615    2773    3236
XMTDAT      001570    698#    751     801    1236#   1256#   1267    1333#   1334#   1335#   1336#   1337    1346#   1358#
                    1363#   1364#   1365#   1366#   1367    1401#   1410#   1441#   1578#   1599#   1602#   1603#   1852#
                    1875#   1876    1878    1883    1884#   1891#   1892    1894#   1909#   1910    1912    1934#   1950#
                    1951    1953    1958    1959#   1966#   1967    1969#   1985#   1986    1988    2020#   2024#   2035#
                    2061#   2068#   2085#   2092#   2109#   2116#   2549#   2562#   2623#   2662#   2700#   2701#   2702
                    2730#   2731#   2732    2772#   2789#   2804#   2829#   2838#   2845    3246#   3277#   3278#   3280
                    3296#   3300    3310#   3338#   3389#   3496#   3501#   3519#
XMTINT      001556    693#   1063#   1260#   2129#   2148#   2156#   2541#
XMTLVL      001560    694#   2155
XMTTST      004622   1234#   2172    2183    2194    2205    2216    2227    2238    2249    2260    2271    2282    2293
                    2304    2315    2326    2337
Y       =   000020    524#   2164    2174#   2175    2185#   2186    2196#   2197    2207#   2208    2218#   2219    2229#
                    2230    2240#   2241    2251#   2252    2262#   2263    2273#   2274    2284#   2285    2295#   2296
                    2306#   2307    2317#   2318    2328#   2329    2339#   2340#   2342    2352#   2353    2363#   2364
```

```
                    2374#  2375   2385#  2386   2396#  2397   2407#  2408   2418#  2419   2429#  2430   2440#
                    2441   2451#  2452   2462#  2463   2473#  2474   2484#  2485   2495#  2496   2506#  2507
                    2517#  2856#  2857   2867#  2868   2878#  2879   2889#  2890   2900#  2901   2911#  2912
                    2922#  2923   2933#  2934   2944#  2945   2955#  2956   2966#  2967   2977#  2978   2988#
                    2989   2999#  3000   3010#  3011   3021#  3022   3032#  3034#  3035   3045#  3046   3056#
                    3057   3067#  3068   3078#  3079   3089#  3090   3100#  3101   3111#  3112   3122#  3123
                    3133#  3134   3144#  3145   3155#  3156   3166#  3167   3177#  3178   3188#  3189   3199#
                    3200   3210#
$CTLU   017666      1209   3705#
$ENDAD  003036       659    908#
$SWREG  017641      3683   3700#
$VALUE  017647      3686   3689   3702#
      = 020332       527#   528    530    532    534    536    538    544    546    548    550    552    554
                     556    558    560    562    564    566    568    570    572    574    576    578    580
                     582    584    586    588    590    592    594    596    598    600    602    604    606
                     608    610    612    614    616    618    620    622    624    626    628    630    632
                     634    636    638    640    642    644    646    648    650    652    654    658#   660#
                     664#   668#   675#   679#   681#   683#   685#   700#   702#   785    788    824    997
                    1142   1242   1307   1352   1354   1383   1386   1456   1477   1558   1561   1590   2618
                    2655   2658   2694   2696   2725   2727   2848   2850   3331   3489   3504   3585   3644#
                    3646#
 BAR    002110      739#   2090*
.BASRE  002114      741#
.BKCSR  002112      740#   2114*
.CSR    002106      738#   1089   2066*

  ABS.  020332  000


ERRORS DETECTED:  0

DSK:CZDMAD,DSK:CZDMAD.SEQ=DSK:CZDMAD.P11
RUN-TIME:  7 11 1 SECONDS
RUN-TIME RATIO:  322/20=15.8
CORE USED:  13K  (25 PAGES)

DOCUMENT PAGES:  85
```