DJ11

DJ11 LOGIC TEST

MD-11-DZDJA-E

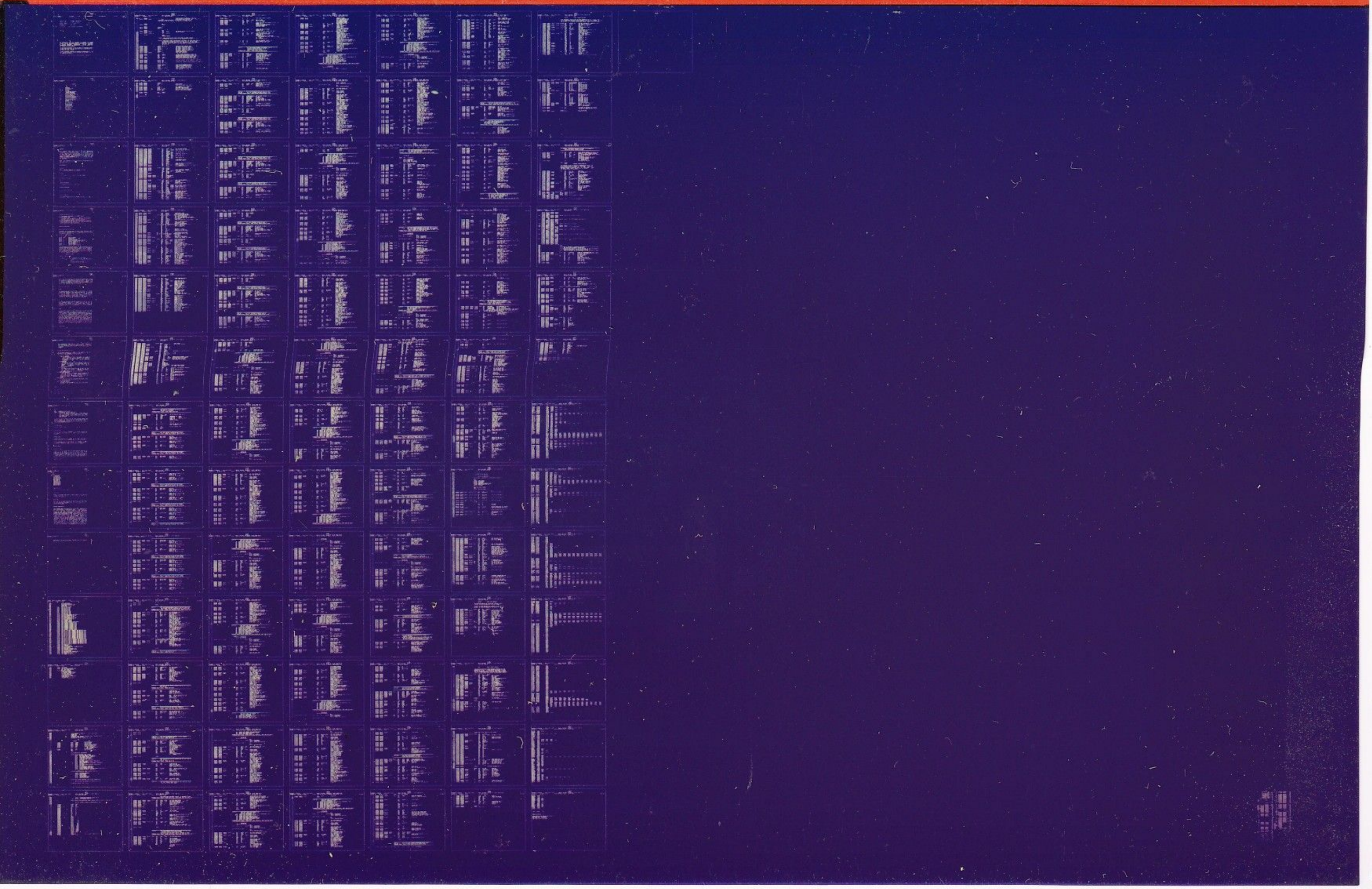
EP-DZDJA-E-DL-B

FICHE 1 OF 1

OCT 1977

digital

MADE IN USA



PRODUCT CODE: MAINDEC-11-DZDJA-E-D

PRODUCT NAME: DJ11 LOGIC TESTS

PROGRAM DATE: MAY 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

CONTENTS

1. ABSTRACT

This program tests the logic of the DJ11 Asyncronous Multiplexer in maintenance mode. It checks that all the control registers function properly, that interrupts occure at the right level, and that data can be transmitted and received correctly. This program does not test that the input and output lead connections are functional. (See MAINDEC-11-DZDJB, PROGRAMS 2 and 3 for on-line testing). The program should be run for at least 2 passes with all switches down.

NOTE: The program will autosize the silo alarm level of the DJ11 and will run at any alarm level set. and will run at any alarm level set.

2. REQUIREMENTS

2.1 Equipment

PDP-11 standard computer with console teletype
Up to 16 DJ11 asyncronous multiplexers.

2.2 Storage

This program uses all of 8K, except ABS LOADER.

2.3 Preliminary Programs

None

3. LOADING PROCEDURE

Use standard procedure for ABS tapes.

4. STARTING PROCEDURE

4.1 Control Switch Settings

See 5.1 (all down for worst case testing)

4.2 Starting Address

The program should always be started at 200. It may be restarted at 1000 after all parameters have been selected.

4.3 Program and Operator Action

1) Load program into memory using ABS loader.
2) Load address 200.
3) If hardware switch register is available, set switches (see sec. 5.1), all down for worst case, press start.
4) If switch-less processor simply press start.
5) Enter parameters (see sec. 5.3) as they are requested.
6) The program will loop and bell will ring once every pass. EOP" is also printed on each pass.
7) A minimum of two passes should always be run.

5. OPERATING PROCEDURE

5.1 Operational Switch Settings

At SA 200, all switches down is worst case testing. Each subtest will be looped upon until completion of 16 passes of that subtest. The bell will ring upon completion of a pass of the entire program.

The switch settings are:

SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... SCOPE LOOP
SW<13> = 1 ..... INHIBIT PRINTOUT
SW<12> = 1 ..... PRINT SILO ALARM LEVEL (IN OCTAL)
SW<11> = 1 ..... INHIBIT ITERATIONS OF SUBTEST
SW<10> = 1 ..... BELL ON ERROR
0 ..... BELL ON PASS COMPLETE
SW<09> = 1 ..... LOOP ON ERROR
SW<08> = 1 ..... LOOP ON TEST IN SW<7:0>

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

(i.e.) 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.

5.2 Subroutine Abstracts

5.2.1 SCOPE

This subroutine call (via a TRAP instruction) is placed between each subtest in the instruction section. It records the starting address of each subtest as it is being entered in location "LAD". If a scope loop is requested, the current subtest will be looped upon. SW<11> on a 1 inhibits iteration of subtests. The contents of "LAD" may be used to determine the last subtest successfully completed.

5.2.2 HLT

This routine (called by an EMT instruction) prints out an error message (see 6.1). If SW<9> is on a 1 and a HLT is executed, the subtest will be looped upon until 16 consecutive good passes are completed. To inhibit typeouts, put SW<13> on a 1. To ring the bell on an error, put SW<10> on a 1.

5.2.4 ALMCK

in the normal operation the "done" bit is set as each character is read into the fi/fo buffer (silo). but this "done" condition can be delayed to cause done on the 5, 9, or 17 character. this is done by cutting one of the jumpers (w1,w2,w3) on the m7285 control board. the program tests for this "silo alarm level" and if sw12 is set (1) it will print out the level at which each dj11 will set "done." the subroutine also adjusts the character counters to ensure that the maximum number of characters to be transfered is a multiple of the silo alarm level. this ensures that all data will be read out of the silo. done will not set if the number of characters in the fi/fo buffer is less than the silo alarm level. (note character present is set on each character in the buffer, regardless of the silo alarm level.)

5.2.5 TRAPCATCHER

A ".+2" - "HALT" sequence is repeated from 0 - 56 to detect

any unexpected traps and a ".+2" - "IOT" sequence is repeated from 60 - 776 to detect any unexpected interrupts. Thus any unexpected traps will HALT at the vector + 2. Any unexpected interrupts will result in an error "HLT" in "IOTRAP".

5.3 Program and Operator Action

The following requests are made to the operator at the beginning of the program. A detailed description of what is required for each parameter is given bellow.

1) "FIRST DJ11 ADDRESS: "
   The CSR address of the first DJ11 you wish to test. Must be between 160000(8) and 177777(8). The default (carriage return) is to 160010(8). "P(REVIOUS)" selects the address previously selected.
2) "VECTOR ADDRESS: "
   The receiver interrupt vector address of the first DJ11 you wish to test. Must be between 300(8) and 1000(8). The default (carriage return) is to 300(8). "P(REVIOUS)" selects the address previously selected.
3) "NO. OF DJ11'S: "
   The number of DJ11 units you wish to test at one time. Must be between 1 and 16. The default (carriage return) is to 1. "P(REVIOUS)" selects the number of units previously selected.
4) "STANDARD CONFIGURATION? "
   "Y(ES)" or default (carriage return) selects 8 level code, no parity. "N(O)" causes requests for code level and parity on all requested lines in groups of four. "P(REVIOUS)" selects the code levels and parities previously selected.
5) "CHAR LENGTH: "
   The code level for the line group specified. Must be 5, 6, 7, or 8. The default (carriage return) is to 8 level code.
6) "PARITY (NO, ODD, EVEN): "
   The type of parity selected for the line group specified. The default (carriage return) is to no parity.

6. ERRORS

6.1 Error Printout

The format is as follows:

ADR DJADR (R1) (R2) (R3) (R4)

Where:

ADR = Address of error HLT
DJADR = CSR address of DJ11 under test
(Rn) = Contents of general register "n". From none to four of these may be typed depending on the number following the HLT; e.g., HLT+3 would type (R1) thru (R3); HLT (by itself) would stop after typing ADR and DJADR.

To find the failing test, look at the listing above the address typed. In most cases the comment beside the HLT tells what was being checked and what was expected. Also, a list of the probable failing logic is given in the comments at the beginning of the test.

6.2 Error Recovery

Restart at 200 or 1000.

6.3 Error Counter

An error count is kept in "ERRORS" . It can be cleared from the console, by restarting at 200, or by reloading the program.

7. RESTRICTIONS

If more than one DJ11 is tested at a time, the device addresses and the vector addresses must all be contiguous.

If this program is run with a monitor, i.e. ACT11 or DDP, the device addresses must follow the floating address convention. DJ11's will be first, starting at 160010,.

8. MISCELLANEOUS

8.1 Execution Time

Due to the verious baud rates available and the ability to check up to 16 DJ11's at once, the execution time can be anywhere from 15 seconds to three and a half hours. The following typical times are for one DJ11 with all lines at the same speed, 8 level code, 2 stop bits, and no parity on a PDP-11/20 without trace trapping. For multiple DJ11's, multiply these times by the number of units selected for test.

BAUD RUN TIME

----	--------
75	00:26:00
110	00:18:00
134.5	00:15:00
150	00:13:00
300	00:06:30
600	00:03:15
1200	00:01:45
1800	00:01:20
2400	00:00:55
4800	00:00:30
9600	00:00:15

8.2 Stack Pointer

Stack is initally set to 1200

8.3 Pass Count

A 32 bit (2 words) pass count is kept in pcnt. It can be cleared from the console, by restarting at 200, or by reloading the program.

8.4 Power Fail

Each test can be power failed with no errors. To use, start the test as usual and power down then up at any time. The routine should type "POWER" and restart the program with no other error typeouts.

9. PROGRAM DESCRIPTION

This program tests the logic of up to 16 DJ11 asynchronous data multiplexers in maintenance mode. It checks that all the control registers function properly, that interrupts occure at the right priority level, and that data can be transmitted and received correctly. The program has many subtests (the code between 2 SCOPE statements) which are run 16 times before continuing to the next. SW<11> on a 1 causes each subtest to be run only once. SW<9> on a 1 enables loop on error. The address "ICNT" contains the iteration count in the left byte and the test number in the right byte. All the subtests should be run sequentially by starting at 200 not by starting at the beginning of the subtest. To loop on a particular subtest, put the test number (see listing) in the right byte of the switch register and SW<8> on a 1. This test will be looped upon until SW<8> is put on a 0 or the right byte is changed. If the test is non-existant, the program will be run as usual. If more than one DJ11 is selected, all the subtests are

performed on one unit at a time. The bell will not ring and EOP print, until a pass has been made on each of the DJ11's selected for test.

```
                                .TITLE MAINDEC-11-DZDJA-E      DJ11 LOGIC TESTS
                                .ENABLE ABS
                                .ENABLE AMA
                                ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                                ;PROGRAM BY KEN CHAPMAN
                                ;MODIFIED BY DAVID L. ADAMS  MAY 1977


                                ;,   SWITCH                    USE
                                ;    ------                    ---------------------
(1)           100000            ;SW15=   100000              ;HALT ON ERROR
(1)           040000            ;SW14=   40000               ;LOOP ON TEST
(1)           020000            ;SW13=   20000               ;INHIBIT ERROR TYPEOUTS
(1)           010000            ;SW12=   10000               ;PRINT SILO ALARM LEVEL
(1)           004000            ;SW11=   4000                ;INHIBIT ITERATIONS
(1)           002000            ;SW10=   2000                ;0 - BELL ON PASS COMPLETE
(1)                                                          ;1 - BELL ON ERROR
(1)           001000            ;SW9=    1000                ;LOOP ON ERROR
(1)           000400            ;SW8=    400                 ;LOOP ON TEST IN SW<7:0>
                                .REM!

                                DJ11 REGISTER BIT ASSIGNMENTS:

                                CONTROL STATUS REGISTER (CSR) XXXXX0
                                        BIT0    RECEIVER ENABLE (READ/WRITE)
                                        BIT1    HALF DUPLEX SELECT (READ/WRITE)
                                        BIT2    MAINTENANCE (READ/WRITE)
                                        BIT3    CLEAR MOS (WRITE ONLY)
                                        BIT4    CLEAR MOS FLAG (READ ONLY)
                                        BIT5    NOT USED
                                        BIT6    RECEIVER INTERUPT ENABLE (READ/WRITE)
                                        BIT7    DONE (READ ONLY)
                                        BIT8    MASTER TRANSMITTER SCAN ENABLE (READ/WRITE)
                                        BIT9    NOT USED
                                        BIT10   READ/WRITE BREAK REGISTER (READ/WRITE)
                                        BIT11   NOT USED
                                        BIT12   STATUS ENABLE (READ/WRITE)
                                        BIT13   FI/FO OVERRUN (READ ONLY)
                                        BIT14   MASTER TRANSMITTER INTERUPT ENABLE (READ/WRITE)
                                        BIT15   TRANSMITTER READY (READ ONLY)

                                RECEIVER BUFFER REGISTER (RBUF) XXXXX2 (READ ONLY)

                                        BIT0-7  RECEIVED CHARACTER
                                        BIT8-11 LINE NUMBER
                                        BIT12   PARITY ERROR
                                        BIT13   FRAMING ERROR
                                        BIT14   UART OVERRUN ERROR
                                        BIT15   CHARACTER PRESENT

                                TRANSMITTER CONTROL REGISTER (TCR) XXXXX4 (READ/WRITE)

                                        BIT0-15 STOP THE SCANNER ON CORRESPONDING LINE

                                TRANSMITTER BUFFER (TBUF) XXXXX6
```

```
                                        BIT0-7  TRANSMITTED CHARACTER (WRITE ONLY)
                                        BIT8-11 LINE NUMBER (READ ONLY)

                                BREAK CONTROL STATUS REGISTER (BCSR) XXXXX4 (BIT10 OF CSR SET) (READ/WRITE)

                                        BIT0-15 TRNSMIT A BREAK ON CORRESPONDING LINE!

          104400                SCOPE=  TRAP
          104000                HLT=    EMT
          000004                TYPE=   IOT
          177776                PS=     177776
          000000                R0=     %0
          000001                R1=     %1
          000002                R2=     %2
          000003                R3=     %3
          000004                R4=     %4
          000005                R5=     %5
          000005                TTY=    %5
          000006                SP=     %6
          000007                PC=     %7
          000007                BELL=   7
          000001                BIT0=       1
          000002                BIT1=       2
          000004                BIT2=       4
          000010                BIT3=      10
          000020                BIT4=      20
          000040                BIT5=      40
          000100                BIT6=     100
          000200                BIT7=     200
          000400                BIT8=     400
          001000                BIT9=    1000
          002000                BIT10=   2000
          004000                BIT11=   4000
          010000                BIT12=  10000
          020000                BIT13=  20000
          040000                BIT14=  40000
          100000                BIT15= 100000
          000340                LEVEL7 = 340
          000000                OPEN =  0
          000020                DJMXNO=16.              ;MAX  ALLOWED NUMBER OF DJ11'S
          001200                STACK   =1200
```

```
                    000000                  .  =    0                       ;TRAP CATCHER IN LOCATIONS 0 THRU 776
(1)  000000  000000  000000                 0,0                             ;LOCATIONS 0 AND 2 CONTAIN "HALT" INSTRUCTIONS
(1)                                         ;LOCATIONS 4 THRU 56 CONTAIN ".+2" AND "HALT" IN EVERY VECTOR
(1)                                         ;LOCATIONS 60 THRU 776 CONTAIN ".+2" AND "IOT" IN EVERY VECTOR
(1)
(1)
(1)
(1)
(1)                 000046                  .  =    46
(1)  000046  015354                         $ENDAD
(1)
(1)                 000174                  .  =    174
(1)  000174  000000                         DISPREG: 0
(1)  000176  000000                         SWREG:   0
(1)
(1)
(1)                 000200                  .  =    200
(1)  000200  000137  001402                 JMP     BEGIN                   ;200 ALWAYS IS THE STARTING ADDRESS
2626                001000                  .  =    1000                    ;RESTART ADDRESS
2627 001000  000137  002330                 JMP     RESTAR
2628
2632                001200                  .  =    1200
2633                                        ADDRESS TABLES
2634                                        ;
2635                                        ;"LENGTH" TABLE IS SET SO THE FIRST FOUR BYTES (2 WORDS) CONTAIN
2636                                        ;       THE CHARACTER LENGTH FOR THE 4 LINE GROUPS OF THE FIRST DJ11, ETC.
2637                                        ;
2638                                        ;"PARITY" TABLE HAS 4 WORDS, ONE FOR EACH LINE GROUP. THE FIRST WORD CONTAINS
2639                                        ;       THE ODD PARITY FLAGS FOR LINES 0-3 OF ALL DJ11'S. THE NEXT WORD
2640                                        ;       HAS PARITY FLAGS FOR LINES 4-7, ETC. EACH BIT IN THE WORD REPRESENTS
2641                                        ;       EACH DJ11. BIT 0 IN EACH OF THE FOUR WORDS IS FOR DJ/0 TO BIT
2642                                        ;       15 FOR DJ/15.
2643
2644 001200  000100                         LENGTH: .BLKB 100               ;TABLE OF CHAR. LENGTHS (MASK)
2645 001300  000000                         PARITY: 0                       ;ODD PARITY FLAGS FOR LINES 0-3
2646 001302  000000                         PARIT2: 0                       ;ODD PARITY FLAGS FOR LINES 4-7
2647 001304  000000                         PARIT3: 0                       ;ODD PARITY FLAGS FOR LINES 10-13
2648 001306  000000                         PARIT4: 0                       ;ODD PARITY FLAGS FOR LINES 14-17
2649 001310  000000                         ICNT:   0                       ;ITERATION COUNT-LH, TEST NO.-RH
2650 001312  000000                         ERRORS: 0                       ;ERROR COUNT REGISTER
2651 001314  000000  000000                 PCNT:   0,0                     ;PASS COUNT REGISTER
2652
2653 001320  160010                         CSR:    160010                  ;CONTROL STATUS REGISTER(DJ UNDER TEST)
2654 001322  160012                         RBUF:   160012                  ;RECEIVER BUFFER REGISTER(DJ UNDER TEST)
2655 001324                                 TCR:                            ;TRANSMITTER CONTROL REGISTER(DJ UNDER TEST)
2656 001324  160014                         BCSR:   160014                  ;BREAK STATUS REGISTER(DJ UNDER TEST)
2657 001326  160016                         TBUF:   160016                  ;TRANSMITTER BUFFER REGISTER(DJ UNDER TEST)
2658
2659 001330  000300                         RCVVEC: 300                     ;RECEIVER INTERUPT VECTOR ADDRESS(DJ UNDER TEST)
2660 001332  000302                         RCVLVL: 302
2661 001334  000304                         XMTVEC: 304                     ;TRANSMITTER INTERUPT VECTOR ADDRESS(DJ UNDER TEST)
2662 001336  000306                         XMTLVL: 306
2663
2664 001340  160010                         DEVADR: 160010                  ;FIRST DEVICE ADDRESS(SELECTED)
2665 001342  000300                         VECADR: 300                     ;FIRST VECTOR ADDRESS(SELECTED)
2666 001344  000001                         UNITS:  1                       ;NUMBER OF UNITS TO BE TESTED
2667 001346  000000                         TIMER:  0                       ;USED TO SAVE RELATIVE TIMES
2668 001350  000000                         ALMFLG: 0                       ;TEST FLAG FOR SILO ALARM LEVEL
2669 001352  000000                         TIMERA: 0                       ;TIME COUNTERS
```

```
 2670  001354  000000                TIMERB: 0
 2671  001356  000000                COUNT:  0                ;CHAR COUNTER
 2672  001360  000000                SUM:    0                ;SUMATION OF ALARM LEVEL CHAR'S
 2673
 2674  001362  000000                DJUUT:  0                ;UNIT NUMBER OF DJ11 UNDER TEST
 2675  001364  000000                DJLEN:  0                ;CHAR MASK TABLE POINTER
 2676  001366  000000                DJPAR:  0                ;UNIT FLAG (FOR ODD PARITY FLAG)
 2677  001370  160010                DEFADR: 160010           ;DEFAULT FIRST DEVICE ADDRESS
 2678  001372  000300                DEFVEC: 300              ;DEFAULT FIRST VECTOR ADDRESS
 2679
  (1)  001374  177570                SWR:    177570
  (1)  001376  177570                DISPLAY:177570
  (1)
 2680
 2681  001400  177777                FTIME:  -1
 2682
 2683
```

```
                                                ;
001402  012706  001200          BEGIN:  MOV     #STACK, SP      ;SET UP STACK POINTER
001406  004737  020124                  JSR     PC,     SUSWRR  ;CHECK FOR SWITCH REGISTER
001412  012700  000020                  MOV     #20,R0
001416  012720  017204                  MOV     #IOTRAP,(R0)+   ;IOT VECTOR (20)
001422  012720  000340                  MOV     #340,   (R0)+
001426  012720  017044                  MOV     #PDOWN$,(R0)+   ;POWER FAIL VECTOR (24)
001432  012720  000340                  MOV     #340,   (R0)+
001436  012720  016074                  MOV     #EMT$,  (R0)+   ;EMT VECTOR (30)
001442  012720  000340                  MOV     #340,   (R0)+
001446  012720  015732                  MOV     #TRAP$, (R0)+   ;TRAP VECTOR (34)
001452  012720  000340                  MOV     #340,   (R0)+
001456  005037  001312                  CLR     ERRORS          ;CLEAR ERROR COUNTER
001462  005037  001314                  CLR     PCNT            ;CLEAR PASS COUNTER
001466  005037  001316                  CLR     PCNT+2
001472  005037  001350                  CLR     ALMFLG
001476  005737  000042                  TST     @#42            ;CHECK FOR ACT11 OR DDP PRESENT
001502  001404                          BEQ     GETADR          ;BRANCH IF NONE
001504  004737  017704                  JSR     PC,     AUTO    ;GO TO SUBROUTINE TO "MAP" DJ11 ON THE SYS
001510  000137  002330                  JMP     RESTAR          ;SKIP OPERATOR ACTION
                                                ;
001514  000004  017633          GETADR: TYPE,   MTITLE
001520  000004  017344                  TYPE,   RETURN
001524  000004  017352                  TYPE,   MSGADR          ;TYPE "FIRST DJ11 ADDRESS"
001530  004537  016316                  JSR     R5,     READIN  ;READ INPUT FROM TTY AND SAVE
001534  001340                          .WORD   DEVADR          ; IN DEVADR
001536  001366                          BNE     GETADR          ;BRANCH IF BAD INPUT
001540  005737  001340                  TST     DEVADR
001544  001003                          BNE     1$
001546  013737  001370  001340          MOV     DEFADR,DEVADR
001554  042737  000007  001340  1$:     BIC     #7,DEVADR
001562  022737  160000  001340          CMP     #160000,DEVADR
001570  101351                          BHI     GETADR
                                                ;
001572  000004  017402          GETVEC: TYPE,   MSGVEC          ;TYPE "VECTOR ADDRESS:"
001576  004537  016316                  JSR     R5,     READIN  ;READ INPUT FROM TTY AND SAVE
001602  001342                          .WORD   VECADR          ;IN VECADR
001604  001372                          BNE     GETVEC          ;BRANCH IF BAD INPUT
001606  005737  001342                  TST     VECADR          ;CHECK FOR CR
001612  001003                          BNE     1$
001614  012737  000300  001342          MOV     #300,   VECADR  ;SET TO FIRST FLOATING VECTOR
001622  042737  000007  001342  1$:     BIC     #7,     VECADR  ;CLEAR TO MODULO 10
001630  022737  000300  001342          CMP     #300,   VECADR  ;CHECK FOR LOW LIMIT
001636  003355                          BGT     GETVEC
001640  022737  001000  001342          CMP     #1000,  VECADR  ;CHECK FOR UPPER LIMIT
001646  003751                          BLE     GETVEC
001650  000004  017426          GETNUM: TYPE,   MSGNUM          ;TYPE"NUMBER OR UNITS:"
001654  004737  016464                  JSR     PC,READ$        ;READ INPUT FROM THE TTY
                                        ; CHECK STRING FOR VALID DIGITS,TERMINATOR,AND 'P'
001660  012702  016612                  MOV     #INPUT,R2       ;POINTS TO INPUT STRING
001664  112201                          MOVB    (R2)+,R1        ;LOAD 1ST CHAR
001666  001434                          BEQ     1$              ;BRANCH IF IMMEDIATE TERMNATOR
001670  122701  000120                  CMPB    #'P,R1          ;CHECK FOR A P
001674  001443                          BEQ     CONFIG          ;CONFIGURE MODE IF SO
001676  120127  000071                  CMPB    R1,#71          ;MUST BE A VALID ASCII NUMBER
001702  101362                          BHI     GETNUM          ;ELSE RETRY
```

```
2744  001704  162701  000060              SUB     #60,R1          ;STRIP ASCII CODE
2745  001710  003757                      BLE     GETNUM          ;SHOULDN'T BE ZERO OR NEG
2746                                ;1ST CHAR IS A VALID DIGIT, 1 THRU 9 - CHECK REST OF STRING
2747  001712  105722                      TSTB    (R2)+           ;2ND CHAR A TERMINATOR?
2748  001714  001423                      BEQ     3$              ;YES - R1 HAS THE FINAL # OF UNITS
2749  001716  105712                      TSTB    (R2)            ;NO - NXT CHAR MUST BE THE TERMINATOR
2750  001720  001353                      BNE     GETNUM          ;ELSE RETRY.
2751  001722  124227  000071              CMPB    -(R2),#71       ;CHECK 2ND CHAR FOR A VALID NUMBER
2752  001726  101350                      BHI     GETNUM
2753  001730  111203                      MOVB    (R2),R3         ;MAKE IT MORE ACCESSIBLE
2754  001732  162703  000060              SUB     #60,R3          ;STRIP ASCII CODE
2755  001736  100744                      BMI     GETNUM          ;SHOULDN'T BE NEGATIVE
2756                                ;BOTH LOW AND HIGH ORDER DIGITS ARE OK - CONVERT DECIMAL VALUE.
2757  001740  010146                      MOV     R1,-(SP)        ;SAVE IT FOR LATER
2758  001742  006301                      ASL     R1              ;MULT HI ORDER BY 8
2759  001744  006301                      ASL     R1
2760  001746  006301                      ASL     R1
2761  001750  006316                      ASL     (SP)            ;MULTIPLY  IT BY 2
2762  001752  062601                      ADD     (SP)+,R1        ;RESULT IS (HI ORD)*10.
2763  001754  060301                      ADD     R3,R1           ;NOW ADD IN THE LOW ORDER
2764  001756  000402                      BR      3$
2765  001760  012701  000001        1$:   MOV     #1,R1           ;LOAD DEFAULT VALUE HERE
2766                                ;R1 HAS THE CONVERTED VALUE - COMPARE AGAINST MAX ALLOWED.
2767  001764  020127  000020        3$:   CMP     R1,#DJMXNO      ;TOO BIG?
2768  001770  101327                      BHI     GETNUM          ;YES - RETRY.
2769  001772  010137  001344              MOV     R1,UNITS        ;VALUE OK - STORE RESULT
2770  001776  012737  000100  001362      MOV     #100,DJUUT      ;"PRIME" UNIT UNDER TEST NUMBER
2771                                ;
2772  002004  000004  017451        CONFIG: TYPE,   MSGCON          ;TYPE "STANDARD CONFIG?"
2773  002010  004737  016464              JSR     PC,     READ$   ;READ INPUT FROM TTY
2774  002014  122737  000120  016612      CMPB    #'P,    INPUT   ;CHECK FOR "P"
2775  002022  001542                      BEQ     RESTAR          ;BRANCH IF PREVIOUS
2776  002024  012700  000044              MOV     #44,    R0      ;SET UP COUNTER
2777  002030  012701  001200              MOV     #LENGTH,R1      ;POINT TO CHAR LEN TABLE
2778  002034  005021                1$:   CLR     (R1)+           ;PUT CHAR MASK FOR 8 IN CHAR TABLE
2779                                                              ;AND CLR PARITY TABLE
2780  002036  005300                      DEC     R0              ;COUNT DOWN
2781  002040  001375                      BNE     1$              ;BRANCH IF NOT DONE
2782  002042  105737  016612              TSTB    INPUT           ;CHECK FOR CR
2783  002046  001530                      BEQ     RESTAR          ;BRANCH IF DEFAULT
2784  002050  122737  000131  016612      CMPB    #131,   INPUT   ;CHECK FOR "Y"
2785  002056  001524                      BEQ     RESTAR          ;BRANCH IF DEFAULT
2786  002060  122737  000116  016612      CMPB    #116,   INPUT   ;CHECK FOR "N"
2787  002066  001346                      BNE     CONFIG          ;BRANCH IF ILLEGAL ENTRY
2788  002070  005000                      CLR     R0              ;CLR UNIT COUNTER
2789  002072  005001                      CLR     R1              ;CLR LINE COUNTER
2790  002074  012702  001200              MOV     #LENGTH,R2      ;SET UP POINTER TO CHAR MASK TABLE
2791  002100  012703  001300              MOV     #PARITY,R3      ;SET UP POINTER TO PARITY TABLE
2792  002104  012704  000001              MOV     #1,R4           ;SET UP MARKER
2793  002110  000004  017505        TYPLIN: TYPE,   MSGLIN          ;TYPE "LINES "
2794  002114  010105                      MOV     R1,TTY          ;TYPE R1 IN OCTAL
 (1)  002116  004737  016672              JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
2795  002122  000004  017516              TYPE,   MSGDAS          ;TYPE A DASH (-)
2796  002126  062701  000003              ADD     #3,     R1      ;LAST LINE IN GROUP OF 4
2797  002132  010105                      MOV     R1,TTY          ;TYPE R1 IN OCTAL
 (1)  002134  004737  016672              JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
```

```
2798   002140  005201                          INC     R1              ;FIRST LINE NEXT GROUP
2799   002142  000004  017522          GETLEN: TYPE,   MSGLEN          ;TYPE "CHAR LENGTH:  "
2800   002146  005037  016610                  CLR     INHRE
2801   002152  004737  016464                  JSR     PC,     READ$   ;READ FROM THE TTY
2802   002156  105737  016612                  TSTB    INPUT           ;CHECK FOR CR (DEFAULT)
2803   002162  001421                          BEQ     GETPAR          ;BRANCH IF DEFAULT
2804   002164  105737  016613                  TSTB    INPUT+1         ;CHECK FOR BAD DATA
2805   002170  001364                          BNE     GETLEN          ;BRANCH IF BAD
2806   002172  162737  000060  016612          SUB     #60,    INPUT   ;CONVERT FROM ASCII
2807   002200  112712  177777                  MOVB    #-1,    (R2)    ;SET UP MASK
2808   002204  1063!2                  2$:     ASLB    (R2)            ;CLEAR MASK BIT BY BIT
2809   002206  103355                          BCC     GETLEN          ;BAD INPUT, > 8
2810   002210  105337  016612                  DECB    INPUT           ;COUNT
2811   002214  001373                          BNE     2$              ;BRANCH IF NOT DONE
2812   002216  132712  000037                  BITB    #37,    (R2)    ;CHECK FOR 5 CHAR
2813   002222  001347                          BNE     GETLEN          ;BAD INPUT, < 5
2814   002224  005202                          INC     R2              ;INC TO NEXT LINE GROUP
2815   002226  000004  017543          GETPAR: TYPE,   MSGPAR          ;TYPE "PARITY (NO, ODD, EVEN):
2816   002232  005037  016610                  CLR     INHRE
2817   002236  004737  016464                  JSR     PC,READ$        ;READ INPUT FROM TYY
2818   002242  005737  016612                  TST     INPUT           ;CHECK FOR CR
2819   002246  001415                          BEQ     3$              ;BRANCH IF DEFAULT
2820   002250  122737  000116  016612          CMPB    #116,INPUT      ;CHECK FOR "N"
2821   002256  001411                          BEQ     3$              ;BRANCH IF "NO"
2822   002260  122737  000105  016612          CMPB    #105,INPUT      ;CHECK FOR "E"
2823   002266  001405                          BEQ     3$              ;BRANCH IF "EVEN"
2824   002270  122737  000117  016612          CMPB    #117,INPUT      ;CHECK FOR "O"
2825   002276  001353                          BNE     GETPAR          ;BRANCH IF NONE
2826   002300  050413                          BIS     R4,     (R3)    ;SET THE ODD PARITY FLAG
2827   002302  005723                  3$:     TST     (R3)+           ;POINT TO THE NEXT PARITY WORD
2828   002304  032701  000017                  BIT     #17,R1          ;CHECK FOR NEXT UNIT
2829   002310  001277                          BNE     TYPLIN          ;BRANCH IF NOT
2830   002312  012703  001300                  MOV     #PARITY,R3      ;RESET PARITY TABLE POINTER
2831   002316  006304                          ASL     R4              ;FLAG TO NEXT UNIT
2832   002320  005200                          INC     R0              ;COUNT UNITS
2833   002322  020037  001344                  CMP     R0,UNITS        ;CHECK FOR LAST
2834   002326  001270                          BNE     TYPLIN          ;BRANCH IF MORE
```

```
2836
2837  002330  000005                    RESTAR: RESET                   ;ISSUE RESET
2838  002332  012706  001200                    MOV     #STACK, SP      ;SET UP STACK POINTER
2839  002336  005737  001400                    TST     FTIME
2840  002342  001410                           BEQ     CLRVEC
2841  002344  022737  000176  001374           CMP     #SWREG, SWR
2842  002352  001004                           BNE     CLRVEC
2843  002354  004737  020204                    JSR     PC,     CNTLU
2844  002360  005037  001400                    CLR     FTIME
2845  002364  012700  000300           CLRVEC: MOV     #300,R0         ;BEGINNING OF FLOATING VECTORS
2846  002370  005720                   1$:     TST     (R0)+
2847  002372  010060  177776                    MOV     R0,-2(R0)       ;".+2"
2848  002376  012720  000004                    MOV     #IOT,   (R0)+   ;"IOT"
2849  002402  022700  001000                    CMP     #1000,R0
2850  002406  001370                           BNE     1$
2851  002410  062737  000010  001320           ADD     #10,    CSR     ;UPDATE DEVICE ADDRESS TO NEXT UNIT
2852  002416  062737  000010  001330           ADD     #10,    RCVVEC  ;UPDATE DEVICE VECTOR ADDRESS
2853  002424  062737  000004  001364           ADD     #4,     DJLEN   ;MOVE CHAR TABLE POINTER
2854  002432  006337  001366                    ASL     DJPAR           ;UPDATE PARITY FLAG MARKER
2855  002436  023737  001362  001344           CMP     DJUUT,UNITS
2856  002444  002416                           BLT     2$
2857  002446  005037  001362                    CLR     DJUUT
2858  002452  013737  001340  001320           MOV     DEVADR,CSR
2859  002460  013737  001342  001330           MOV     VECADR,RCVVEC
2860  002466  012737  001200  001364           MOV     #LENGTH,DJLEN   ;INIT CHAR TABLE POINTER
2861  002474  012737  000001  001366           MOV     #1,DJPAR        ;INIT PARITY FLAG MARKER
2862  002502  005237  001362           2$:     INC     DJUUT
2863  002506  013701  001320                    MOV     CSR,    R1      ;SET UP ALL THE REGISTER ADDRESSES
2864  002512  062701  000002                    ADD     #2,     R1      ;ADD 2
2865  002516  010137  001322                    MOV     R1,     RBUF    ;SET UP RECEIVER BUFFER
2866  002522  062701  000002                    ADD     #2,     R1      ;ADD 2
2867  002526  010137  001324                    MOV     R1,     TCR     ;SET UP TRANSMITTER CONTROL REG
2868                                                                    ; AND BREAK STATUS REG
2869  002532  062701  000002                    ADD     #2,     R1      ;ADD 2
2870  002536  010137  001326                    MOV     R1,     TBUF    ;SET UP TRANSMITTER BUFFER
2871  002542  013701  001330                    MOV     RCVVEC, R1      ;POINTER FOR VECTOR SETUP
2872  002546  005721                           TST     (R1)+           ;INC R1
2873  002550  010137  001332                    MOV     R1,     RCVLVL  ;SET INT LVL
2874  002554  005721                           TST     (R1)+           ;INC R1
2875  002556  010137  001334                    MOV     R1,     XMTVEC  ;TRANSMITTER VECTOR
2876  002562  005721                           TST     (R1)+           ;INC R1
2877  002564  010137  001336                    MOV     R1,     XMTLVL  ;XMT INT LVL ADR
2878  002570  005037  001310                    CLR     ICNT
2879  002574  005037  016070                    CLR     LAD
2880  002600  104400                           SCOPE
2881
2882
```

```
4560
 (1)                                ;*****************************************************************************
 (1)                                ; TEST 1:       TEST ABILITY TO REFERENCE CSR WITHOUT TRAPPING
 (1)                                ;               AND ABILITY TO CLR CSR.
 (1)                                ;PROBABLE FAULTY LOGIC:  M105; M7285 (D2-6) E29
 (1)                                ;*****************************************************************************
 (1)
 (1)  002602  012737  002640  000004  TST1:  MOV   #1$,    @#4     ;SET UP TIME-OUT TRAP VECTOR
 (1)  002610  012737  000340  000006        MOV   #LEVEL7,@#6
 (1)  002616  005777  176476                TST   @CSR              ;REFERENCE CSR  (READ)
 (1)  002622  005077  176472                CLR   @CSR              ;CLEAR CSR      (WRITE)
 (1)  002626  005577  176466                ADC   @CSR              ;CHECK CSR      (READ AND WRITE)
 (1)  002632  001405                        BEQ   2$                ;BRANCH IF OK
 (1)  002634  104000                        HLT                     ;CSR NOT CLEARED
 (1)
 (1)  002636  000403                        BR    2$                ;SKIP ISR
 (1)
 (1)  002640  012601                1$:     MOV   (SP)+,R1          ;SAVE RTI ADR FOR TYPING
 (1)  002642  104001                        HLT+1                   ;CAN'T REFERENCE CSR!
 (1)
 (1)  002644  005726                        TST   (SP)+             ;FINISH CLEARING STACK
 (1)  002646  012737  000006  000004  2$:   MOV   #6,@#4
 (1)  002654  005037  000006               CLR   @#6
 (1)  002660  104400                        SCOPE
4561
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 2:        TEST THAT CSR BIT1 CAN BE SET AND CLEARED
 (1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E25,E7, (D2-4) E47,E64
 (1)                                ;*****************************************************************************
 (1)
 (1)  002662  012777  000002  176430  TST2:  MOV   #BIT1,@CSR       ;SET BIT1
 (1)  002670  032777  000002  176422        BIT   #BIT1,@CSR       ;CHECK THAT BIT1 IS SET
 (1)  002676  001001                        BNE   .+4               ;BRANCH IF OK
 (1)  002700  104000                        HLT                     ;CSR BIT1 FAILED TO SET
 (1)
 (1)  002702  032777  177775  176410        BIT   #177775,@CSR     ;CHECK THAT NO OTHER BIT SET
 (1)  002710  001401                        BEQ   .+4               ;BRANCH IF OK
 (1)  002712  104000                        HLT                     ;EXTRA BIT SET IN CSR
 (1)
 (1)  002714  005077  176400                CLR   @CSR              ;CLEAR BIT1
 (1)  002720  032777  000002  176372        BIT   #BIT1,@CSR       ;CHECK THAT BIT1 IS CLEARED
 (1)  002726  001401                        BEQ   .+4               ;BRANCH IF OK
 (1)  002730  104000                        HLT                     ;CSR BIT1 FAILED TO CLEAR
 (1)
 (1)  002732  104400                        SCOPE
4562
 (1)                                ;*****************************************************************************
 (1)                                ;TEST 3:        TEST THAT CSR BIT2 CAN BE SET AND CLEARED
 (1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E25,E7, (D2-4) E47,E64
 (1)                                ;*****************************************************************************
 (1)
 (1)  002734  012777  000004  176356  TST3:  MOV   #BIT2,@CSR       ;SET BIT2
 (1)  002742  032777  000004  176350        BIT   #BIT2,@CSR       ;CHECK THAT BIT2 IS SET
 (1)  002750  001001                        BNE   .+4               ;BRANCH IF OK
 (1)  002752  104000                        HLT                     ;CSR BIT2 FAILED TO SET
 (1)
```

```
   (1)  002754  032777  177773  176336          BIT     #177773,@CSR    ;CHECK THAT NO OTHER BIT SET
   (1)  002762  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  002764  104000                          HLT                     ;EXTRA BIT SET IN CSR
   (1)
   (1)  002766  005077  176326                  CLR     @CSR            ;CLEAR BIT2
   (1)  002772  032777  000004  176320          BIT     #BIT2,@CSR      ;CHECK THAT BIT2 IS CLEARED
   (1)  003000  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003002  104000                          HLT                     ;CSR BIT2 FAILED TO CLEAR
   (1)
   (1)  003004  104400                          SCOPE
 4563
   (1)                            ;*************************************************************************
   (1)                            ;TEST 4:        TEST THAT CSR BIT6 CAN BE SET AND CLEARED
   (1)                            ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E4,E18, (D2-4) E47,E64
   (1)                            ;*************************************************************************
   (1)
   (1)  003006  012777  000100  176304  TST4:   MOV     #BIT6,@CSR      ;SET BIT6
   (1)  003014  032777  000100  176276          BIT     #BIT6,@CSR      ;CHECK THAT BIT6 IS SET
   (1)  003022  001001                          BNE     .+4             ;BRANCH IF OK
   (1)  003024  104000                          HLT                     ;CSR BIT6 FAILED TO SET
   (1)
   (1)  003026  032777  177677  176264          BIT     #177677,@CSR    ;CHECK THAT NO OTHER BIT SET
   (1)  003034  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003036  104000                          HLT                     ;EXTRA BIT SET IN CSR
   (1)
   (1)  003040  005077  176254                  CLR     @CSR            ;CLEAR BIT6
   (1)  003044  032777  000100  176246          BIT     #BIT6,@CSR      ;CHECK THAT BIT6 IS CLEARED
   (1)  003052  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003054  104000                          HLT                     ;CSR BIT6 FAILED TO CLEAR
   (1)
   (1)  003056  104400                          SCOPE
 4564
   (1)                            ;*************************************************************************
   (1)                            ;TEST 5:        TEST THAT CSR BIT8 CAN BE SET AND CLEARED
   (1)                            ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E6,E18, (D2-4) E47,E31
   (1)                            ;*************************************************************************
   (1)
   (1)  003060  012777  000400  176232  TST5:   MOV     #BIT8,@CSR      ;SET BIT8
   (1)  003066  032777  000400  176224          BIT     #BIT8,@CSR      ;CHECK THAT BIT8 IS SET
   (1)  003074  001001                          BNE     .+4             ;BRANCH IF OK
   (1)  003076  104000                          HLT                     ;CSR BIT8 FAILED TO SET
   (1)
   (1)  003100  032777  177377  176212          BIT     #177377,@CSR    ;CHECK THAT NO OTHER BIT SET
   (1)  003106  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003110  104000                          HLT                     ;EXTRA BIT SET IN CSR
   (1)
   (1)  003112  005077  176202                  CLR     @CSR            ;CLEAR BIT8
   (1)  003116  032777  000400  176174          BIT     #BIT8,@CSR      ;CHECK THAT BIT8 IS CLEARED
   (1)  003124  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003126  104000                          HLT                     ;CSR BIT8 FAILED TO CLEAR
   (1)
   (1)  003130  104400                          SCOPE
 4565
   (1)                            ;*************************************************************************
   (1)                            ;TEST 6:        TEST THAT CSR BIT10 CAN BE SET AND CLEARED
   (1)                            ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E30,E24, (D2-4) E47,E31
```

```
(1)                                     ;*****************************************************************************
(1)
(1)  003132  012777  002000  176160  TST6:   MOV     #BIT10,@CSR     ;SET BIT10
(1)  003140  032777  002000  176152          BIT     #BIT10,@CSR     ;CHECK THAT BIT10 IS SET
(1)  003146  001001                          BNE     .+4             ;BRANCH IF OK
(1)  003150  104000                          HLT                     ;CSR BIT10 FAILED TO SET
(1)
(1)  003152  032777  175777  176140          BIT     #175777,@CSR    ;CHECK THAT NO OTHER BIT SET
(1)  003160  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  003162  104000                          HLT                     ;EXTRA BIT SET IN CSR
(1)
(1)  003164  005077  176130                  CLR     @CSR            ;CLEAR BIT10
(1)  003170  032777  002000  176122          BIT     #BIT10,@CSR     ;CHECK THAT BIT10 IS CLEARED
(1)  003176  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  003200  104000                          HLT                     ;CSR BIT10 FAILED TO CLEAR
(1)
(1)  003202  104400                          SCOPE
4566
(1)                                     ;*****************************************************************************
(1)                                     ;TEST 7:        TEST THAT CSR BIT12 CAN BE SET AND CLEARED
(1)                                     ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5,E1, (D2-4) E47,E31
(1)                                     ;*****************************************************************************
(1)
(1)  003204  012777  010000  176106  TST7:   MOV     #BIT12,@CSR     ;SET BIT12
(1)  003212  032777  010000  176100          BIT     #BIT12,@CSR     ;CHECK THAT BIT12 IS SET
(1)  003220  001001                          BNE     .+4             ;BRANCH IF OK
(1)  003222  104000                          HLT                     ;CSR BIT12 FAILED TO SET
(1)
(1)  003224  032777  167777  176066          BIT     #167777,@CSR    ;CHECK THAT NO OTHER BIT SET
(1)  003232  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  003234  104000                          HLT                     ;EXTRA BIT SET IN CSR
(1)
(1)  003236  005077  176056                  CLR     @CSR            ;CLEAR BIT12
(1)  003242  032777  010000  176050          BIT     #BIT12,@CSR     ;CHECK THAT BIT12 IS CLEARED
(1)  003250  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  003252  104000                          HLT                     ;CSR BIT12 FAILED TO CLEAR
(1)
(1)  003254  104400                          SCOPE
4567
(1)                                     ;*****************************************************************************
(1)                                     ;TEST 10:       TEST THAT CSR BIT14 CAN BE SET AND CLEARED
(1)                                     ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E3,E1, (D2-4) E47,E31
(1)                                     ;*****************************************************************************
(1)
(1)  003256  012777  040000  176034  TST10:  MOV     #BIT14,@CSR     ;SET BIT14
(1)  003264  032777  040000  176026          BIT     #BIT14,@CSR     ;CHECK THAT BIT14 IS SET
(1)  003272  001001                          BNE     .+4             ;BRANCH IF OK
(1)  003274  104000                          HLT                     ;CSR BIT14 FAILED TO SET
(1)
(1)  003276  032777  137777  176014          BIT     #137777,@CSR    ;CHECK THAT NO OTHER BIT SET
(1)  003304  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  003306  104000                          HLT                     ;EXTRA BIT SET IN CSR
(1)
(1)  003310  005077  176004                  CLR     @CSR            ;CLEAR BIT14
(1)  003314  032777  040000  175776          BIT     #BIT14,@CSR     ;CHECK THAT BIT14 IS CLEARED
(1)  003322  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
(1)  003324  104000                        HLT                     ;CSR BIT14 FAILED TO CLEAR
(1)
(1)  003326  104400                        SCOPE
4568
(1)                                ;*************************************************************************
(1)                                ;TEST 11:       TEST THAT RECEIVER ENABLE (BIT0 OF THE CSR) CAN BE SET
(1)                                ;                 AND CLEARED, AND THAT CLEAR MOS (BIT3) IS WRITE ONLY.
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E26,E36,E7,E24, (D2-8) E17,E14,E15
(1)                                ;*************************************************************************
(1)
(1)  003330  012777  000004  175762  TST11:  MOV     #BIT2,  @CSR    ;SET MAINTENANCE MODE (BIT2)
(1)  003336  005005                        CLR     R5              ;SET UP COUNTER
(1)  003340  052777  000010  175752        BIS     #BIT3,  @CSR    ;SET CLEAR MOS (BIT3)
(1)  003346  017701  175746        1$:     MOV     @CSR,   R1      ;SAVE CSR
(1)  003352  032701  000010                BIT     #BIT3,  R1      ;CHECK CLEAR MOS (BIT3)
(1)  003356  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  003360  104001                        HLT+1                   ;CLEAR MOS (BIT3) SET (WRITE-ONLY)
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  003362  032701  000020                BIT     #BIT4,  R1      ;CHECK CLEAR MOS FLAG
(1)  003366  001403                        BEQ     2$              ;BRANCH IF CLEARED
(1)  003370  105305                        DECB    R5              ;WAIT FOR MOS TO CLEAR
(1)  003372  001365                        BNE     1$              ;BRANCH IF MORE TIME
(1)  003374  104001                        HLT+1                   ;CLEAR MOS FLAG (BIT4) FAILED TO CLEAR
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  003376  022701  000004        2$:     CMP     #BIT2,  R1      ;CHECK THAT ONLY MAINTENANCE BIT SET
(1)  003402  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  003404  104001                        HLT+1                   ;CLEAR MOS CLEARED MAINTENANCE
(1)                                                                ; OR SET OTHER CSR BITS
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  003406  052777  000001  175704        BIS     #BIT0,  @CSR    ;SET RECEIVER ENABLE
(1)  003414  017701  175700                MOV     @CSR,   R1      ;SAVE CSR
(1)  003420  032777  000001  175672        BIT     #BIT0,  @CSR    ;CHECK THAT RECEIVER ENABLE SET
(1)  003426  001001                        BNE     .+4             ;BRANCH IF OK
(1)  003430  104001                        HLT+1                   ;RECEIVER ENABLE FAILED TO SET
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  003432  022777  000005  175660        CMP     #5,     @CSR    ;CHECK REST OF CSR
(1)  003440  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  003442  104001                        HLT+1                   ;CSR ERROR
(1)                                                                ;R1 = CONTENTS OF CSR
(1)                                ;NOTE:  IF THE TTY MODULE IS BEING USED AND DONE (BIT7) IS SET,
(1)                                ;       THE ERROR COULD BE DUE TO MAINTENANCE OR CLEAR MOS NOT WORKING.
(1)  003444  042777  000001  175646        BIC     #BIT0,  @CSR    ;CLEAR RECEIVER ENABLE
(1)  003452  017701  175642                MOV     @CSR,   R1      ;SAVE CSR
(1)  003456  022777  000004  175634        CMP     #BIT2,  @CSR    ;CHECK CSR
(1)  003464  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  003466  104001                        HLT+1                   ;RECEIVER ENABLE DIDN'T CLEAR
(1)                                                                ; OR OTHER CSR BIT SET
(1)                                                                ;R1 = CONTENTS OF CSR
(1)
(1)  003470  104400                        SCOPE
4569
(1)                                ;*************************************************************************
(1)                                ; TEST 12:      TEST THAT CSR RESPONDS PROPERLY TO BYTE COMMANDS
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-4) E47
(1)                                ;*************************************************************************
(1)
```

```
   (1)  003472  012777  052506  175620  TST12:  MOV     #052506,@CSR    ;SET TEST NUMBER IN CSR
   (1)  003500  105077  175614                  CLRB    @CSR            ;CLR EVEN BYTE
   (1)  003504  017701  175610                  MOV     @CSR,   R1      ;SAVE CSR
   (1)  003510  022701  052400                  CMP     #052400,R1      ;CHECK CSR
   (1)  003514  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003516  104001                          HLT+1                   ;EVEN BYTE CLR FAILED ON CSR
   (1)                                                                  ;R1 = CONTENTS OF CSR
   (1)  003520  012777  052506  175572          MOV     #052506,@CSR    ;SET TEST NUMBER IN CSR
   (1)  003526  005237  001320                  INC     CSR             ;INC TO ODD BYTE
   (1)  003532  105077  175562                  CLRB    @CSR            ;CLR ODD BYTE
   (1)  003536  005337  001320                  DEC     CSR             ;RESTORE TO EVEN
   (1)  003542  017701  175552                  MOV     @CSR,   R1      ;SAVE CSR
   (1)  003546  022701  000106                  CMP     #000106,R1      ;CHECK CSR
   (1)  003552  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003554  104001                          HLT+1                   ;ODD BYTE CLR FAILED ON CSR
   (1)                                                                  ;R1 = CONTENTS OF CSR
   (1)
   (1)  003556  104400                          SCOPE
 4570
   (1)                          ;*************************************************************************
   (1)                          ;TEST 13:       TEST THAT THE BIS AND BIC INSTRUCTIONS SET AND CLEAR R/W
   (1)                          ;               BITS OF CSR
   (1)                          ;PROBABLE FAULTY LOGIC:  M7285 (D2-4) E47
   (1)                          ;*************************************************************************
   (1)
   (1)  003560  005077  175534  TST13:  CLR     @CSR            ;CLEAR THE CSR
   (1)  003564  017701  175530          MOV     @CSR,   R1      ;CHECK AND SAVE CSR
   (1)  003570  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
   (1)  003572  104001                  HLT+1                   ;RESET FAILED TO CLR CSR
   (1)
   (1)  003574  052777  052506  175516  BIS     #052506,        @CSR    ;SET ALL R/W BITS OF CSR
   (1)  003602  022777  052506  175510  CMP     #052506,        @CSR    ;CHECK THAT THEY GOT SET
   (1)  003610  001401                  BEQ     .+4             ;BRANCH IF OK
   (1)  003612  104000                  HLT                     ;REG FAILED CMP
   (1)
   (1)  003614  042777  052506  175476  BIC     #052506,        @CSR    ;CLEAR CSR
   (1)  003622  017701  175472          MOV     @CSR,   R1      ;CHECK AND SAVE CSR
   (1)  003626  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
   (1)  003630  104001                  HLT+1                   ;CLR FAILED TO CLR CSR
   (1)
   (1)  003632  104400                  SCOPE
 4571
   (1)                          ;*************************************************************************
   (1)                          ;TEST 14:       TEST BITS OF TCR FOR READ/WRITE CAPABILITY
   (1)                          ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) ALL, (D2-3) E8,E20,E21,E43,E41
   (1)                          ;*************************************************************************
   (1)
   (1)  003634  012777  177777  175462  TST14:  MOV     #177777,@TCR    ;SET ALL BITS OF TCR
   (1)  003642  017701  175456                  MOV     @TCR,   R1      ;CHECK AND SAVE TCR
   (1)  003646  022701  177777                  CMP     #177777,R1      ;CHECK THAT ALL THE BITS ARE SET
   (1)  003652  001401                          BEQ     .+4             ;BRANCH IF OK
   (1)  003654  104001                          HLT+1                   ;BIT(S) OF TCR FAILED TO SET
   (1)
   (1)  003656  005077  175442                  CLR     @TCR            ;CLEAR TCR
   (1)  003662  017701  175436                  MOV     @TCR,   R1      ;CHECK THAT IT CLEARED AND SAVE
   (1)  003666  001401                          BEQ     .+4             ;BRANCH IF CLR
```

```
  (1)  003670  104001                        HLT+1                   ;BIT(S) OF TCR FAILED TO CLEAR
  (1)
  (1)  003672  104400                        SCOPE
 4572
  (1)                              ;*************************************************************************
  (1)                              ; TEST 15:      TEST THAT TCR RESPONDS PROPERLY TO BYTE COMMANDS
  (1)                              ;PROBABLE FAULTY LOGIC:  M7285 (D2-3) E41
  (1)                              ;*************************************************************************
  (1)
  (1)  003674  012777  177777  175422  TST15:  MOV     #177777,@TCR    ;SET TEST NUMBER IN TCR
  (1)  003702  105077  175416                  CLRB    @TCR            ;CLR EVEN BYTE
  (1)  003706  017701  175412                  MOV     @TCR,   R1      ;SAVE TCR
  (1)  003712  022701  177400                  CMP     #177400,R1      ;CHECK TCR
  (1)  003716  001401                          BEQ     .+4             ;BRANCH IF OK
  (1)  003720  104001                          HLT+1                   ;EVEN BYTE CLR FAILED ON TCR
  (1)                                                                  ;R1 = CONTENTS OF TCR
  (1)  003722  012777  177777  175374          MOV     #177777,@TCR    ;SET TEST NUMBER IN TCR
  (1)  003730  005237  001324                  INC     TCR             ;INC TO ODD BYTE
  (1)  003734  105077  175364                  CLRB    @TCR            ;CLR ODD BYTE
  (1)  003740  005337  001324                  DEC     TCR             ;RESTORE TO EVEN
  (1)  003744  017701  175354                  MOV     @TCR,   R1      ;SAVE TCR
  (1)  003750  022701  000377                  CMP     #000377,R1      ;CHECK TCR
  (1)  003754  001401                          BEQ     .+4             ;BRANCH IF OK
  (1)  003756  104001                          HLT+1                   ;ODD BYTE CLR FAILED ON TCR
  (1)                                                                  ;R1 = CONTENTS OF TCR
  (1)
  (1)  003760  104400                        SCOPE
 4573
  (1)                              ;*************************************************************************
  (1)                              ;TEST 16:       TEST THAT THE BIS AND BIC INSTRUCTIONS SET AND CLEAR R/W
  (1)                              ;               BITS OF TCR
  (1)                              ;PROBABLE FAULTY LOGIC:  M7285 (D2-3) E41
  (1)                              ;*************************************************************************
  (1)
  (1)  003762  005077  175336  TST16:  CLR     @TCR            ;CLEAR THE TCR
  (1)  003766  017701  175332          MOV     @TCR,   R1      ;CHECK AND SAVE TCR
  (1)  003772  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
  (1)  003774  104001                  HLT+1                   ;RESET FAILED TO CLR TCR
  (1)
  (1)  003776  052777  177777  175320  BIS     #177777,        @TCR    ;SET ALL R/W BITS OF TCR
  (1)  004004  022777  177777  175312  CMP     #177777,        @TCR    ;CHECK THAT THEY GOT SET
  (1)  004012  001401                  BEQ     .+4             ;BRANCH IF OK
  (1)  004014  104000                  HLT                     ;REG FAILED CMP
  (1)
  (1)  004016  042777  177777  175300  BIC     #177777,        @TCR    ;CLEAR TCR
  (1)  004024  017701  175274          MOV     @TCR,   R1      ;CHECK AND SAVE TCR
  (1)  004030  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
  (1)  004032  104001                  HLT+1                   ;CLR FAILED TO CLR TCR
  (1)
  (1)  004034  104400                  SCOPE
 4574  004036  012737  000001  001346  MOV     #1,     TIMER   ;INITIALIZE TIMER
 4575  004044  012737  004052  016070  MOV     #.+6,   LAD     ;RESET LOOP ADDRESS
 4576
  (1)                              ;*************************************************************************
  (1)                              ;TEST 17:       TEST THAT TRANSMIT READY (BIT15 OF CSR) SETS AND CLEARS
  (1)                              ;                 WHEN TCRO IS SET AND CLEARED.
```

```
(1)                                 ;                ALSO CHECK THAT THE RIGHT LINE NO. (0) APPEARS IN TBUFF.
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D2-5) ALL, (D2-6) E23, E32, E33, E49, (D2-2) E3, E1
(1)                                 ;*****************************************************************************
(1)
(1)  004052  012777  000400  175240 TST17:  MOV     #400,   @CSR    ;SET XMTR SCAN ENABLE
(1)  004060  052777  000001  175236         BIS     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE 0
(1)  004066  005000                         CLR     R0              ;SET UP WAIT COUNTER
(1)  004070  017701  175224          1$:    MOV     @CSR,   R1      ;CHECK AND SAVE XMTR READY
(1)  004074  100404                         BMI     2$              ;BRANCH IF SET OK
(1)  004076  005200                         INC     R0              ;WAIT A WHILE
(1)  004100  001373                         BNE     1$
(1)  004102  104001                         HLT+1                   ;XMTR READY FAILED TO SET
(1)
(1)  004104  000416                         BR      3$              ;SKIP LINE # CHECK ON ERROR
(1)
(1)  004106  050037  001346          2$:    BIS     R0,     TIMER   ;SET UP TIMER
(1)  004112  017701  175210                 MOV     @TBUF,  R1      ;SAVE LINE NUMBER
(1)  004116  105001                         CLRB    R1
(1)  004120  000301                         SWAB    R1
(1)  004122  022701  000000                 CMP     #0,     R1      ;CHECK THAT THE XMTR STOPPED ON LINE 0
(1)  004126  001401                         BEQ     .+4             ;BRANCH IF OK
(1)  004130  104001                         HLT+1                   ;WRONG LINE NUMBER APPEARED IN TBUF
(1)
(1)  004132  017702  175162                 MOV     @CSR,   R2      ;CHECK AND SAVE XMTR READY
(1)  004136  100401                         BMI     .+4             ;BRANCH IF OK
(1)  004140  104002                         HLT+2                   ;READING THE TBUF CLEARED XMTR READY
(1)
(1)  004142  042777  000001  175154  3$:    BIC     #BIT0,  @TCR    ;CLR XMTR CONTROL BIT, LINE 0
(1)  004150  017701  175144                 MOV     @CSR,   R1      ;CHECK AND SAVE XMTR READY
(1)  004154  100001                         BPL     .+4             ;BRANCH IF OK
(1)  004156  104001                         HLT+1                   ;XMTR READY FAILED TO CLEAR WHEN
(1)                                                                 ;   XMTR CONTROL FOR LINE 0 WAS CLEARED
(1)
(1)  004160  104400                         SCOPE
(1)
4577
(1)                                 ;*****************************************************************************
(1)                                 ;TEST 20:        TEST THAT TRANSMIT READY (BIT15 OF CSR) SETS AND CLEARS
(1)                                 ;                  WHEN EACH TCR BIT IS SET AND CLEARED.
(1)                                 ;
(1)                                 ;                ALSO CHECK THAT THE RIGHT LINE NO.  APPEARS IN TBUFF.
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D2-5) ALL, (D2-6) E23, E32, E33, E49, (D2-2) E3, E1
(1)                                 ;*****************************************************************************
(1)
(1)  004162  012777  000400  175130 TST20:  MOV     #400,   @CSR    ;SET XMTR SCAN ENABLE
(1)  004170  005001                         CLR     R1
(1)  004172  012704  000001                 MOV     #1,     R4      ;SET UP MARKER
(1)  004176  050477  175122          4$:    BIS     R4,     @TCR    ;SET XMTR CONTROL BIT, EACH LINE
(1)  004202  005000                         CLR     R0              ;SET UP WAIT COUNTER
(1)  004204  017702  175110          1$:    MOV     @CSR,   R2      ;CHECK AND SAVE XMTR READY
(1)  004210  100404                         BMI     2$              ;BRANCH IF SET OK
(1)  004212  005300                         DEC     R0              ;WAIT A WHILE
(1)  004214  001373                         BNE     1$
(1)  004216  104002                         HLT+2                   ;XMTR READY FAILED TO SET
(1)                                                                 ;R1=LINE #
(1)                                                                 ;R2=CSR
(1)
```

```
(1)  004220  000414                          BR      3$              ;SKIP LINE # CHECK ON ERROR
(1)
(1)  004222  050037  001346        2$:       BIS     R0,     TIMER   ;SET UP TIMER
(1)  004226  017702  175074                  MOV     @TBUF,  R2      ;SAVE LINE NUMBER
(1)  004232  000302                          SWAB    R2
(1)  004234  020201                          CMP     R2,     R1      ;CHECK THAT THE XMTR STOPPED ON RIGHT LINE
(1)  004236  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  004240  104002                          HLT+2                   ;WRONG LINE NUMBER APPEARED IN TBUF
(1)                                                                  ;R1=LINE #(SHOULD BE)
(1)                                                                  ;R2=LINE # (TBUF)
(1)
(1)
(1)  004242  017703  175052                  MOV     @CSR,   R3      ;CHECK AND SAVE XMTR READY
(1)  004246  100401                          BMI     .+4             ;BRANCH IF OK
(1)  004250  104003                          HLT+3                   ;READING THE TBUF CLEARED XMTR READY
(1)
(1)  004252  040477  175046        3$:       BIC     R4,     @TCR    ;CLR XMTR CONTROL BIT, EACH LINE
(1)  004256  017702  175036                  MOV     @CSR,   R2      ;CHECK AND SAVE XMTR READY
(1)  004262  100001                          BPL     .+4             ;BRANCH IF OK
(1)  004264  104002                          HLT+2                   ;XMTR READY FAILED TO CLEAR WHEN
(1)                                                                  ;    XMTR CONTROL FOR LINE .LINE WAS CLEARED
(1)                                                                  ;R1 = LINE #
(1)                                                                  ;R2 = CONTENTS OF CSR
(1)
(1)  004266  005201                          INC     R1              ;INC LINE COUNTER TO NEXT LINE
(1)  004270  006304                          ASL     R4              ;SHIFT MARKER TO NEXT LINE
(1)  004272  103341                          BCC     4$              ;BRANCH IF NOT LAST LINE
(1)
(1)  004274  104400                          SCOPE
(1)
4578
(1)                                  ;*************************************************************************
(1)                                  ;TEST 21:       TEST THAT MASTER TRANSMIT SCAN ENABLE (CSR BIT 8) ON A 0
(1)                                  ;                DISABLES TRANSMITTER READY (CSR BIT 15)
(1)                                  ;                WHEN TCR BIT, LINE 0 IS SET.
(1)                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-6) E49,E23,E32,E33
(1)                                  ;*************************************************************************
(1)
(1)  004276  005077  175016  175014  TST21:  CLR     @CSR            ;CLEAR CONTROL STATUS
(1)  004302  052777  000001          TST21:  BIS     #BIT0,@TCR      ;SET XMTR CONTROL BIT, LINE 0
(1)  004310  013700  001346                  MOV     TIMER,R0        ;GET TIMER FROM PREVIOUS TEST
(1)  004314  017701  175000          1$:     MOV     @CSR,R1         ;CHECK AND SAVE XMTR READY
(1)  004320  100002                          BPL     2$              ;BRANCH IF NOT SET
(1)  004322  104001                          HLT+1                   ;XMTR READY SET WITHOUT
(1)  004324  000402                          BR      3$              ;MASTER TRANS SCAN ENABLE
(1)  004326  005300                  2$:     DEC     R0              ;WAIT A WHILE
(1)  004330  001371                          BNE     1$              ;AND CHECK AGAIN
(1)
(1)  004332  052777  000400  174760  3$:     BIS     #400,@CSR       ;SET MASTER TRAN SCAN ENABLE
(1)  004340  005000                          CLR     R0
(1)  004342  017701  174752          4$:     MOV     @CSR,   R1      ;CHECK AND SAVE XMTR READY
(1)  004346  100403                          BMI     5$
(1)  004350  005200                          INC     R0
(1)  004352  001373                          BNE     4$
(1)  004354  104001                          HLT+1                   ;TRAN READY NEVER CAME UP
(1)                                                                  ;TCR BIT WAS SET FIRST
```

```
(1)                                                             ;THEN MASTER TRAN SCAN ENABLE
(1)  004356  042777  000001  174740  5$:     BIC     #BIT0,  @TCR    ;CLEAR IT OUT
(1)  004364  017701  174730                  MOV     @CSR,   R1      ;CHECK AND SAVE TRAN READY
(1)  004370  100001                          BPL     .+4             ;BRANCH IF OK
(1)  004372  104001                          HLT+1                   ;TRAN READY DIDN'T CLEAR.
(1)
(1)  004374  104400                          SCOPE
(1)
4579
(1)                                     ;*************************************************************************
(1)                                     ;TEST 22:       TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 7
(1)                                     ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                     ;*************************************************************************
(1)
(1)  004376  012777  004452  174730  TST22:  MOV     #ISR22,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  004404  012777  000340  174724          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  004412  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  004420  052737  000340  177776          BIS     #340,@#PS       ;SET PS TO LEVEL 7
(1)  004426  012777  040400  174664          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  004434  012777  000001  174662          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  004442  017701  174652          1$:     MOV     @CSR,   R1      ;WAIT
(1)  004446  100375                          BPL     1$
(1)  004450  000404                          BR      END22           ;OK, BRANCH IF NO INTERRUPT
(1)
(1)  004452  104000                  ISR22:  HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 7
(1)  004454  012716  004462                  MOV     #END22,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  004460  000002                          RTI
(1)
(1)  004462  013777  001336  174644  END22:  MOV     XMTLVL,@XMTVEC
(1)  004470  012777  000004  174640          MOV     #IOT,   @XMTLVL
(1)  004476  005077  174622                  CLR     @TCR
(1)  004502  005077  174612                  CLR     @CSR
(1)  004506  042737  000340  177776          BIC     #340,@#PS
(1)
(1)  004514  104400                          SCOPE
(1)
4580
(1)                                     ;*************************************************************************
(1)                                     ;TEST 23:       TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 6
(1)                                     ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                     ;*************************************************************************
(1)
(1)  004516  012777  004572  174610  TST23:  MOV     #ISR23,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  004524  012777  000340  174604          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  004532  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  004540  052737  000300  177776          BIS     #300,@#PS       ;SET PS TO LEVEL 6
(1)  004546  012777  040400  174544          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  004554  012777  000001  174542          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  004562  017701  174532          1$:     MOV     @CSR,   R1      ;WAIT
(1)  004566  100375                          BPL     1$
(1)  004570  000404                          BR      END23           ;OK, BRANCH IF NO INTERRUPT
(1)
(1)  004572  104000                  ISR23:  HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 6
(1)  004574  012716  004602                  MOV     #END23,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  004600  000002                          RTI
(1)
```

```
(1)  004602  013777  001336  174524  END23:  MOV     XMTLVL,@XMTVEC
(1)  004610  012777  000004  174520          MOV     #IOT,   @XMTLVL
(1)  004616  005077  174502                  CLR     @TCR
(1)  004622  005077  174472                  CLR     @CSR
(1)  004626  042737  000340  177776          BIC     #340,@#PS
(1)
(1)  004634  104400                          SCOPE
(1)
4581
(1)                                ;*************************************************************************
(1)                                ;TEST 24:       TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 5
(1)                                ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                ;*************************************************************************
(1)
(1)  004636  012777  004712  174470  TST24:  MOV     #ISR24,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  004644  012777  000340  174464          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  004652  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  004660  052737  000240  177776          BIS     #240,@#PS       ;SET PS TO LEVEL 5
(1)  004666  012777  040400  174424          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  004674  012777  000001  174422          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  004702  017701  174412          1$:     MOV     @CSR,   R1      ;WAIT
(1)  004706  100375                          BPL     1$
(1)  004710  000404                          BR      END24           ;OK, BRANCH IF NO INTERRUPT
(1)
(1)  004712  104000                  ISR24:  HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 5
(1)  004714  012716  004722                  MOV     #END24,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  004720  000002                          RTI
(1)
(1)  004722  013777  001336  174404  END24:  MOV     XMTLVL,@XMTVEC
(1)  004730  012777  000004  174400          MOV     #IOT,   @XMTLVL
(1)  004736  005077  174362                  CLR     @TCR
(1)  004742  005077  174352                  CLR     @CSR
(1)  004746  042737  000340  177776          BIC     #340,@#PS
(1)
(1)  004754  104400                          SCOPE
(1)
4582
(1)                                ;*************************************************************************
(1)                                ;TEST 25:       TEST THAT INTERRUPT OCCURS AT LEVEL 4
(1)                                ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                ;*************************************************************************
(1)
(1)  004756  012777  005034  174350  TST25:  MOV     #ISR25,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  004764  012777  000340  174344          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  004772  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  005000  052737  000200  177776          BIS     #200,@#PS       ;SET PS TO LEVEL 4
(1)  005006  012777  040400  174304          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  005014  012777  000001  174302          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  005022  017701  174272          1$:     MOV     @CSR,   R1      ;WAIT
(1)  005026  100375                          BPL     1$
(1)  005030  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 4
(1)  005032  000403                          BR      END25           ;CONTINUE
(1)
(1)  005034  012716  005042          ISR25:  MOV     #END25,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  005040  000002                          RTI
(1)
```

```
(1)  005042  013777  001336  174264  END25:  MOV     XMTLVL,@XMTVEC
(1)  005050  012777  000004  174260          MOV     #IOT,   @XMTLVL
(1)  005056  005077  174242                  CLR     @TCR
(1)  005062  005077  174232                  CLR     @CSR
(1)  005066  042737  000340  177776          BIC     #340,@#PS
(1)
(1)  005074  104400                          SCOPE
(1)
4583
(1)                                 ;*************************************************************************
(1)                                 ;TEST 26:       TEST THAT INTERRUPT OCCURS AT LEVEL 3
(1)                                 ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                 ;*************************************************************************
(1)
(1)  005076  012777  005154  174230  TST26:  MOV     #ISR26,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  005104  012777  000340  174224          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  005112  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  005120  052737  000140  177776          BIS     #140,@#PS       ;SET PS TO LEVEL 3
(1)  005126  012777  040400  174164          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  005134  012777  000001  174162          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  005142  017701  174152          1$:     MOV     @CSR,   R1      ;WAIT
(1)  005146  100375                          BPL     1$
(1)  005150  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 3
(1)  005152  000403                          BR      END26           ;CONTINUE
(1)
(1)  005154  012716  005162          ISR26:  MOV     #END26,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  005160  000002                          RTI
(1)
(1)  005162  013777  001336  174144  END26:  MOV     XMTLVL,@XMTVEC
(1)  005170  012777  000004  174140          MOV     #IOT,   @XMTLVL
(1)  005176  005077  174122                  CLR     @TCR
(1)  005202  005077  174112                  CLR     @CSR
(1)  005206  042737  000340  177776          BIC     #340,@#PS
(1)
(1)  005214  104400                          SCOPE
(1)
4584
(1)                                 ;*************************************************************************
(1)                                 ;TEST 27:       TEST THAT INTERRUPT OCCURS AT LEVEL 2
(1)                                 ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                 ;*************************************************************************
(1)
(1)  005216  012777  005274  174110  TST27:  MOV     #ISR27,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  005224  012777  000340  174104          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  005232  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  005240  052737  000100  177776          BIS     #100,@#PS       ;SET PS TO LEVEL 2
(1)  005246  012777  040400  174044          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  005254  012777  000001  174042          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  005262  017701  174032          1$:     MOV     @CSR,   R1      ;WAIT
(1)  005266  100375                          BPL     1$
(1)  005270  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 2
(1)  005272  000403                          BR      END27           ;CONTINUE
(1)
(1)  005274  012716  005302          ISR27:  MOV     #END27,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  005300  000002                          RTI
(1)
```

```
(1)  005302  013777  001336  174024  END27:  MOV     XMTLVL,@XMTVEC
(1)  005310  012777  000004  174020          MOV     #IOT,   @XMTLVL
(1)  005316  005077  174002                  CLR     @TCR
(1)  005322  005077  173772                  CLR     @CSR
(1)  005326  042737  000340  177776          BIC     #340,@#PS
(1)
(1)  005334  104400                          SCOPE
(1)
4585
(1)                                 ;*************************************************************************
(1)                                 ;TEST 30:       TEST THAT INTERRUPT OCCURS AT LEVEL 1
(1)                                 ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                 ;*************************************************************************
(1)
(1)  005336  012777  005414  173770  TST30:  MOV     #ISR30,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  005344  012777  000340  173764          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  005352  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  005360  052737  000040  177776          BIS     #040,@#PS       ;SET PS TO LEVEL 1
(1)  005366  012777  040400  173724          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  005374  012777  000001  173722          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  005402  017701  173712          1$:     MOV     @CSR,   R1      ;WAIT
(1)  005406  100375                          BPL     1$
(1)  005410  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 1
(1)  005412  000403                          BR      END30           ;CONTINUE
(1)
(1)  005414  012716  005422          ISR30:  MOV     #END30,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  005420  000002                          RTI
(1)
(1)  005422  013777  001336  173704  END30:  MOV     XMTLVL,@XMTVEC
(1)  005430  012777  000004  173700          MOV     #IOT,   @XMTLVL
(1)  005436  005077  173662                  CLR     @TCR
(1)  005442  005077  173652                  CLR     @CSR
(1)  005446  042737  000340  177776          BIC     #340,@#PS
(1)
(1)  005454  104400                          SCOPE
(1)
4586
(1)                                 ;*************************************************************************
(1)                                 ;TEST 31:       TEST THAT INTERRUPT OCCURS AT LEVEL 0
(1)                                 ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                 ;*************************************************************************
(1)
(1)  005456  012777  005534  173650  TST31:  MOV     #ISR31,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
(1)  005464  012777  000340  173644          MOV     #340,@XMTLVL    ;AT LEVEL 7
(1)  005472  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
(1)  005500  052737  000000  177776          BIS     #000,@#PS       ;SET PS TO LEVEL 0
(1)  005506  012777  040400  173604          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
(1)  005514  012777  000001  173602          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)  005522  017701  173572          1$:     MOV     @CSR,   R1      ;WAIT
(1)  005526  100375                          BPL     1$
(1)  005530  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 0
(1)  005532  000403                          BR      END31           ;CONTINUE
(1)
(1)  005534  012716  005542          ISR31:  MOV     #END31,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  005540  000002                          RTI
(1)
```

```
   (1)  005542  013777  001336  173564  END31:  MOV     XMTLVL,@XMTVEC
   (1)  005550  012777  000004  173560          MOV     #IOT,   @XMTLVL
   (1)  005556  005077  173542                  CLR     @TCR
   (1)  005562  005077  173532                  CLR     @CSR
   (1)  005566  042737  000340  177776          BIC     #340,@#PS
   (1)
   (1)  005574  104400                          SCOPE
   (1)
  4587  005576  005037  001346                  CLR     TIMER           ;INITIALIZE TIMER
  4588  005602  012737  005610  016070          MOV     #.+6,   LAD     ;RESET LOOP ADDRESS
  4589
   (1)                                  ;*************************************************************************
   (1)                                  ;TEST 32:       TEST THAT LINE 0 CAN TRANSMIT AND
   (1)                                  ;               RECEIVE A CHARACTER. (377)
   (1)                                  ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
   (1)                                  ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
   (1)                                  ;       4$:  CHECKS THAT RIGHT LINE # (0) IN FI/FO
   (1)                                  ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
   (1)                                  ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
   (1)                                  ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
   (1)                                  ;       8$:  CHECKS THAT DONE CLEARS
   (1)                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
   (1)                                  ;*************************************************************************
   (1)
   (1)                                  ;               INITIALIZE
   (1)                                  ;       DEVICE"CSR"REGISTER
   (1)                                  ;                                               SET:
   (1)  005610  004737  015420          TST32:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
   (1)                                                                  ;BIT3 = CLEAR MOS
   (1)                                                                  ;BIT8 = MASTER XMTR SCAN ENB
   (1)                                  ;WAIT FOR MOS TO CLEAR
   (1)                                  ;                               SET:
   (1)                                  ;                               ;BIT0 = RECEIVER ENABLE
   (1)                                  ;
   (1)  005614  052777  000001  173502          BIS     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE 0
   (1)  005622  017701  173472          LOP32:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
   (1)  005626  100375                          BPL     LOP32
   (1)  005630  012777  000377  173470          MOV     #377,   @TBUF   ;SEND A RUBOUT
   (1)  005636  005000                          CLR     R0              ;CLEAR COUNTER
   (1)  005640  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
   (1)  005642  001376                          BNE     1$
   (1)  005644  017701  173452                  MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
   (1)  005650  100405                          BMI     3$              ;BRANCH WHEN FOUND
   (1)  005652  005200                          INC     R0              ;TIME COUNTER
   (1)  005654  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
   (1)  005656  017702  173436                  MOV     @CSR,   R2      ;SAVE CSR
   (1)  005662  104002                          HLT+2                   ;CHARACTER READY DIDN'T SET
   (1)                                                                  ;R1 = CONTENTS OF RBUF
   (1)                                                                  ;R2 = CONTENTS OF CSR
   (1)  005664  050037  001346          3$:     BIS     R0,TIMER        ;SAVE THE TIMER
   (1)  005670  032701  070000                  BIT     #70000, R1      ;CHECK FOR ERROR BITS
   (1)  005674  001401                          BEQ     .+4             ;BRANCH IF NONE
   (1)  005676  104001                          HLT+1                   ;ERROR IN RECEIVED CHAR
   (1)                                                                  ;R1 = CONTENTS OF RBUF
   (1)                                                                  ;BIT14=UART OVERRUN
   (1)                                                                  ;BIT13=FRAMING ERROR
```

```
(1)
(1)  005700  010102                  4$:   MOV    R1,     R2      ;BIT12=PARITY ERROR
(1)  005702  042702  170377               BIC    #170377,R2      ;DUPLICATE DATA WORD
(1)  005706  000302                       SWAB   R2              ;MASK LINE#
(1)  005710  122702  000000               CMPB   #0.,    R2      ;LINE # IN LOW BYTE
(1)  005714  001401                       BEQ    .+4             ;CHECK LINE #
(1)  005716  104001                       HLT+1                  ;BRANCH IF OK
(1)                                                              ;WRONG LINE # RECEIVED
(1)                                                              ;R1 = CONTENTS OF RBUF
(1)                                                              ;BITS8-11 = LINE #
(1)  005720  117702  173440         5$:   MOVB   @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  005724  130201                       BITB   R2,     R1      ;CHECK CHAR LENGTH.
(1)  005726  001401                       BEQ    .+4             ;BRANCH IF OK
(1)  005730  104002                       HLT+2                  ;WRONG CHARACTER LENGTH
(1)                                                              ;R1=DATA FROM FI/FO
(1)                                                              ;R2=MASK (BITS SET NOT EXPECTED)
(1)  005732  105102                 6$:   COMB   R2              ;REVERSE THE MASK
(1)  005734  120102                       CMPB   R1,     R2      ;CHECK THE ACTURAL DATA
(1)  005736  001401                       BEQ    .+4             ;BRANCH IF OK
(1)  005740  104002                       HLT+2                  ;WRONG CHAR LEN OR DATA ERROR
(1)                                                              ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                              ;R2=DATA (LOW BYTE) EXPECTED
(1)  005742  017701  173354         7$:   MOV    @RBUF,  R1      ;READ FI/FO
(1)  005746  100001                       BPL    .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  005750  104001                       HLT+1                  ;CHARACTER PRESENT STAYED SET
(1)                                                              ;R1 = CONTENTS OF RBUF
(1)  005752  017701  173342         8$:   MOV    @CSR,   R1      ;SAVE THE CSR
(1)  005756  022701  100405               CMP    #100405,R1      ;CHECK THE CSR
(1)  005762  001401                       BEQ    .+4             ;BRANCH IF OK
(1)  005764  104001                       HLT+1                  ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                              ;R1 = CONTENTS OF CSR
(1)  005766  005077  173332               CLR    @TCR            ;CLEAR TCR
(1)  005772  005077  173322               CLR    @CSR            ;CLEAR CSR
(1)  005776  104400                       SCOPE
4590
(1)                                ;*****************************************************************************
(1)                                ;TEST 33:        TEST THAT LINE 1 CAN TRANSMIT AND
(1)                                ;                RECEIVE A CHARACTER. (377)
(1)                                ;        1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                ;        3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                ;        4$:  CHECKS THAT RIGHT LINE # (1) IN FI/FO
(1)                                ;        5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                ;        6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                ;        7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                ;        8$:  CHECKS THAT DONE CLEARS
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                ;*****************************************************************************
(1)
(1)                                ;        INITIALIZE
(1)                                ;     DEVICE"CSR"REGISTER
(1)                                                        SET:
(1)  006000  004737  015420         TST33: JSR    PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                              ;BIT3 = CLEAR MOS
(1)                                                              ;BIT8 = MASTER XMTR SCAN ENB
(1)
(1)                                ;WAIT FOR MOS TO CLEAR
(1)                                ;                       SET:
(1)                                ;                       ;BIT0 = RECEIVER ENABLE
```

```
(1)                                        ;
(1)  006004  052777  000002  173312        BIS     #BIT1,  @TCR    ;SET XMTR CONTROL BIT, LINE 1
(1)  006012  017701  173302        LOP33:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  006016  100375                        BPL     LOP33
(1)  006020  012777  000377  173300        MOV     #377,   @TBUF   ;SEND A RUBOUT
(1)  006026  005000                        CLR     R0              ;CLEAR COUNTER
(1)  006030  105305                1$:     DECB    R5              ;SHORT WAIT LOOP
(1)  006032  001376                        BNE     1$
(1)  006034  017701  173262                MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  006040  100405                        BMI     3$              ;BRANCH WHEN FOUND
(1)  006042  005200                        INC     R0              ;TIME COUNTER
(1)  006044  001371                        BNE     1$              ;BRANCH IF NOT TIME-OUT
(1)  006046  017702  173246                MOV     @CSR,   R2      ;SAVE CSR
(1)  006052  104002                        HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)                                                                ;R2 = CONTENTS OF CSR
(1)  006054  050037  001346        3$:     BIS     R0,TIMER        ;SAVE THE TIMER
(1)  006060  032701  070000                BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1)  006064  001401                        BEQ     .+4             ;BRANCH IF NONE
(1)  006066  104001                        HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)                                                                ;BIT14=UART OVERRUN
(1)                                                                ;BIT13=FRAMING ERROR
(1)                                                                ;BIT12=PARITY ERROR
(1)  006070  010102                4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  006072  042702  170377                BIC     #170377,R2      ;MASK LINE#
(1)  006076  000302                        SWAB    R2              ;LINE # IN LOW BYTE
(1)  006100  122702  000001                CMPB    #1.,    R2      ;CHECK LINE #
(1)  006104  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  006106  104001                        HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)                                                                ;BITS8-11 = LINE #
(1)  006110  117702  173250        5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  006114  130201                        BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  006116  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  006120  104002                        HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                                ;R1=DATA FROM FI/FO
(1)                                                                ;R2=MASK (BITS SET NOT EXPECTED)
(1)  006122  105102                6$:     COMB    R2              ;REVERSE THE MASK
(1)  006124  120102                        CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  006126  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  006130  104002                        HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                ;R2=DATA (LOW BYTE) EXPECTED
(1)  006132  017701  173164        7$:     MOV     @RBUF,  R1      ;READ FI/FO
(1)  006136  100001                        BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  006140  104001                        HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)  006142  017701  173152        8$:     MOV     @CSR,   R1      ;SAVE THE CSR
(1)  006146  022701  100405                CMP     #100405,R1      ;CHECK THE CSR
(1)  006152  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  006154  104001                        HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  006156  005077  173142                CLR     @TCR            ;CLEAR TCR
(1)  006162  005077  173132                CLR     @CSR            ;CLEAR CSR
(1)  006166  104400                        SCOPE
```

```
4591
 (1)                    ;*****************************************************************************
 (1)                    ;TEST 34:       TEST THAT LINE 2 CAN TRANSMIT AND
 (1)                    ;               RECEIVE A CHARACTER. (377)
 (1)                    ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
 (1)                    ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
 (1)                    ;       4$:  CHECKS THAT RIGHT LINE # (2) IN FI/FO
 (1)                    ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
 (1)                    ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
 (1)                    ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
 (1)                    ;       8$:  CHECKS THAT DONE CLEARS
 (1)                    ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
 (1)                    ;*****************************************************************************
 (1)
 (1)                    ;               INITIALIZE
 (1)                    ;       DEVICE"CSR"REGISTER
 (1)                    ;                                               SET:
 (1)  006170  004737  015420    TST34:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
 (1)                                                            ;BIT3 = CLEAR MOS
 (1)                                                            ;BIT8 = MASTER XMTR SCAN ENB
 (1)                    ;WAIT FOR MOS TO CLEAR
 (1)                    ;                                       SET:
 (1)                    ;                                       ;BIT0 = RECEIVER ENABLE
 (1)                    ;
 (1)  006174  052777  000004  173122    BIS     #BIT2,  @TCR    ;SET XMTR CONTROL BIT, LINE 2
 (1)  006202  017701  173112    LOP34:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 (1)  006206  100375            BPL     LOP34
 (1)  006210  012777  000377  173110    MOV     #377,   @TBUF   ;SEND A RUBOUT
 (1)  006216  005000            CLR     R0              ;CLEAR COUNTER
 (1)  006220  105305            1$:     DECB    R5              ;SHORT WAIT LOOP
 (1)  006222  001376            BNE     1$
 (1)  006224  017701  173072    MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
 (1)  006230  100405            BMI     3$              ;BRANCH WHEN FOUND
 (1)  006232  005200            INC     R0              ;TIME COUNTER
 (1)  006234  001371            BNE     1$              ;BRANCH IF NOT TIME-OUT
 (1)  006236  017702  173056    MOV     @CSR,   R2      ;SAVE CSR
 (1)  006242  104002            HLT+2                   ;CHARACTER READY DIDN'T SET
 (1)                                                    ;R1 = CONTENTS OF RBUF
 (1)                                                    ;R2 = CONTENTS OF CSR
 (1)  006244  050037  001346    3$:     BIS     R0,TIMER        ;SAVE THE TIMER
 (1)  006250  032701  070000            BIT     #70000, R1      ;CHECK FOR ERROR BITS
 (1)  006254  001401            BEQ     .+4             ;BRANCH IF NONE
 (1)  006256  104001            HLT+1                   ;ERROR IN RECEIVED CHAR
 (1)                                                    ;R1 = CONTENTS OF RBUF
 (1)                                                    ;BIT14=UART OVERRUN
 (1)                                                    ;BIT13=FRAMING ERROR
 (1)                                                    ;BIT12=PARITY ERROR
 (1)  006260  010102            4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
 (1)  006262  042702  170377            BIC     #170377,R2      ;MASK LINE#
 (1)  006266  000302            SWAB    R2              ;LINE # IN LOW BYTE
 (1)  006270  122702  000002            CMPB    #2.,    R2      ;CHECK LINE #
 (1)  006274  001401            BEQ     .+4             ;BRANCH IF OK
 (1)  006276  104001            HLT+1                   ;WRONG LINE # RECEIVED
 (1)                                                    ;R1 = CONTENTS OF RBUF
 (1)                                                    ;BITS8-11 = LINE #
 (1)  006300  117702  173060    5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
```

```
(1)  006304  130201                        BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  006306  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  006310  104002                        HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                                ;R1=DATA FROM FI/FO
(1)                                                                ;R2=MASK (BITS SET NOT EXPECTED)
(1)  006312  105102                 6$:    COMB    R2              ;REVERSE THE MASK
(1)  006314  120102                        CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  006316  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  006320  104002                        HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                ;R2=DATA (LOW BYTE) EXPECTED
(1)  006322  017701  172774         7$:    MOV     @RBUF,  R1      ;READ FI/FO
(1)  006326  100001                        BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  006330  104001                        HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)  006332  017701  172762         8$:    MOV     @CSR,   R1      ;SAVE THE CSR
(1)  006336  022701  100405                CMP     #100405,R1      ;CHECK THE CSR
(1)  006342  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  006344  104001                        HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  006346  005077  172752                CLR     @TCR            ;CLEAR TCR
(1)  006352  005077  172742                CLR     @CSR            ;CLEAR CSR
(1)  006356  104400                        SCOPE
4592
(1)                                 ;*****************************************************************************
(1)                                 ;TEST 35:       TEST THAT LINE 3 CAN TRANSMIT AND
(1)                                 ;               RECEIVE A CHARACTER. (377)
(1)                                 ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                 ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                 ;       4$:  CHECKS THAT RIGHT LINE # (3) IN FI/FO
(1)                                 ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                 ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                 ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                 ;       8$:  CHECKS THAT DONE CLEARS
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                 ;*****************************************************************************
(1)
(1)                                 ;               INITIALIZE
(1)                                 ;       DEVICE"CSR"REGISTER
(1)                                 ;                                               SET:
(1)  006360  004737  015420         TST35: JSR      PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                 ;                               ;BIT3 = CLEAR MOS
(1)                                 ;                               ;BIT8 = MASTER XMTR SCAN ENB
(1)                                 ;WAIT FOR MOS TO CLEAR
(1)                                 ;                               SET:
(1)                                 ;                               ;BIT0 = RECEIVER ENABLE
(1)                                 ;
(1)  006364  052777  000010  172732        BIS     #BIT3,  @TCR    ;SET XMTR CONTROL BIT, LINE 3
(1)  006372  017701  172722         LOP35: MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  006376  100375                        BPL     LOP35
(1)  006400  012777  000377  172720        MOV     #377,   @TBUF   ;SEND A RUBOUT
(1)  006406  005000                        CLR     R0              ;CLEAR COUNTER
(1)  006410  105305                 1$:    DECB    R5              ;SHORT WAIT LOOP
(1)  006412  001376                        BNE     1$
(1)  006414  017701  172702                MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  006420  100405                        BMI     3$              ;BRANCH WHEN FOUND
```

```
(1)  006422  005200                    INC     R0              ;TIME COUNTER
(1)  006424  001371                    BNE     1$              ;BRANCH IF NOT TIME-OUT
(1)  006426  017702  172666            MOV     @CSR,   R2      ;SAVE CSR
(1)  006432  104002                    HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)                                                            ;R2 = CONTENTS OF CSR
(1)  006434  050037  001346    3$:     BIS     R0,TIMER        ;SAVE THE TIMER
(1)  006440  032701  070000            BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1)  006444  001401                    BEQ     .+4             ;BRANCH IF NONE
(1)  006446  104001                    HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)                                                            ;BIT14=UART OVERRUN
(1)                                                            ;BIT13=FRAMING ERROR
(1)                                                            ;BIT12=PARITY ERROR
(1)  006450  010102    4$:     MOV     R1,     R2              ;DUPLICATE DATA WORD
(1)  006452  042702  170377            BIC     #170377,R2      ;MASK LINE#
(1)  006456  000302                    SWAB    R2              ;LINE # IN LOW BYTE
(1)  006460  122702  000003            CMPB    #3.,    R2      ;CHECK LINE #
(1)  006464  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  006466  104001                    HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)                                                            ;BITS8-11 = LINE #
(1)  006470  117702  172670    5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  006474  130201                    BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  006476  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  006500  104002                    HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                            ;R1=DATA FROM FI/FO
(1)                                                            ;R2=MASK (BITS SET NOT EXPECTED)
(1)  006502  105102    6$:     COMB    R2                      ;REVERSE THE MASK
(1)  006504  120102            CMPB    R1,     R2              ;CHECK THE ACTURAL DATA
(1)  006506  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  006510  104002                    HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                            ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                            ;R2=DATA (LOW BYTE) EXPECTED
(1)  006512  017701  172604    7$:     MOV     @RBUF,  R1      ;READ FI/FO
(1)  006516  100001                    BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  006520  104001                    HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)  006522  017701  172572    8$:     MOV     @CSR,   R1      ;SAVE THE CSR
(1)  006526  022701  100405            CMP     #100405,R1      ;CHECK THE CSR
(1)  006532  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  006534  104001                    HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                            ;R1 = CONTENTS OF CSR
(1)  006536  005077  172562            CLR     @TCR            ;CLEAR TCR
(1)  006542  005077  172552            CLR     @CSR            ;CLEAR CSR
(1)  006546  104400                    SCOPE
4593 006550  005237  001364            INC     DJLEN
4594 006554  012737  006562  016070    MOV     #.+6,   LAD
4595
(1)                            ;*****************************************************************************
(1)                            ;TEST 36:       TEST THAT LINE 4 CAN TRANSMIT AND
(1)                            ;               RECEIVE A CHARACTER. (377)
(1)                            ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                            ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                            ;       4$:  CHECKS THAT RIGHT LINE # (4) IN FI/FO
(1)                            ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
```

```
(1)                                     ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                     ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                     ;       8$:  CHECKS THAT DONE CLEARS
(1)                                     ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                     ;*************************************************************************
(1)
(1)                                     ;               INITIALIZE
(1)                                     ;       DEVICE"CSR"REGISTER
(1)                                     ;                                                       SET:
(1)  006562  004737  015420             TST36:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                                     ;BIT3 = CLEAR MOS
(1)                                                                     ;BIT8 = MASTER XMTR SCAN ENB
(1)                                     ;WAIT FOR MOS TO CLEAR
(1)                                     ;                                                       SET:
(1)                                     ;                                                       ;BIT0 = RECEIVER ENABLE
(1)                                     ;
(1)  006566  052777  000020  172530             BIS     #BIT4,  @TCR    ;SET XMTR CONTROL BIT, LINE 4
(1)  006574  017701  172520             LOP36:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  006600  100375                             BPL     LOP36
(1)  006602  012777  000377  172516             MOV     #377,   @TBUF   ;SEND A RUBOUT
(1)  006610  005000                             CLR     R0              ;CLEAR COUNTER
(1)  006612  105305                     1$:     DECB    R5              ;SHORT WAIT LOOP
(1)  006614  001376                             BNE     1$
(1)  006616  017701  172500                     MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  006622  100405                             BMI     3$              ;BRANCH WHEN FOUND
(1)  006624  005200                             INC     R0              ;TIME COUNTER
(1)  006626  001371                             BNE     1$              ;BRANCH IF NOT TIME-OUT
(1)  006630  017702  172464                     MOV     @CSR,   R2      ;SAVE CSR
(1)  006634  104002                             HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;R2 = CONTENTS OF CSR
(1)  006636  050037  001346             3$:     BIS     R0,TIMER        ;SAVE THE TIMER
(1)  006642  032701  070000                     BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1)  006646  001401                             BEQ     .+4             ;BRANCH IF NONE
(1)  006650  104001                             HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BIT14=UART OVERRUN
(1)                                                                     ;BIT13=FRAMING ERROR
(1)                                                                     ;BIT12=PARITY ERROR
(1)  006652  010102                     4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  006654  042702  170377                     BIC     #170377,R2      ;MASK LINE#
(1)  006660  000302                             SWAB    R2              ;LINE # IN LOW BYTE
(1)  006662  122702  000004                     CMPB    #4.,    R2      ;CHECK LINE #
(1)  006666  001401                             BEQ     .+4             ;BRANCH IF OK
(1)  006670  104001                             HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BITS8-11 = LINE #
(1)  006672  117702  172466             5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  006676  130201                             BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  006700  001401                             BEQ     .+4             ;BRANCH IF OK
(1)  006702  104002                             HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                                     ;R1=DATA FROM FI/FO
(1)                                                                     ;R2=MASK (BITS SET NOT EXPECTED)
(1)  006704  105102                     6$:     COMB    R2              ;REVERSE THE MASK
(1)  006706  120102                             CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  006710  001401                             BEQ     .+4             ;BRANCH IF OK
```

```
(1)  006712  104002                        HLT+2                    ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                   ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                   ;R2=DATA (LOW BYTE) EXPECTED
(1)  006714  017701  172402          7$:   MOV    @RBUF,  R1         ;READ FI/FO
(1)  006720  100001                        BPL    .+4                ;BRANCH IF CHAR PRESENT NOT SET
(1)  006722  104001                        HLT+1                     ;CHARACTER PRESENT STAYED SET
(1)                                                                   ;R1 = CONTENTS OF RBUF
(1)  006724  017701  172370          8$:   MOV    @CSR,   R1         ;SAVE THE CSR
(1)  006730  022701  100405                CMP    #100405,R1         ;CHECK THE CSR
(1)  006734  001401                        BEQ    .+4                ;BRANCH IF OK
(1)  006736  104001                        HLT+1                     ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                   ;R1 = CONTENTS OF CSR
(1)  006740  005077  172360                CLR    @TCR               ;CLEAR TCR
(1)  006744  005077  172350                CLR    @CSR               ;CLEAR CSR
(1)  006750  104400                        SCOPE
 4596
(1)                                  ;*************************************************************************
(1)                                  ;TEST 37:       TEST THAT LINE 5 CAN TRANSMIT AND
(1)                                  ;               RECEIVE A CHARACTER. (377)
(1)                                  ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                  ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                  ;       4$:  CHECKS THAT RIGHT LINE # (5) IN FI/FO
(1)                                  ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                  ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                  ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                  ;       8$:  CHECKS THAT DONE CLEARS
(1)                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                  ;*************************************************************************
(1)
(1)                                  ;               INITIALIZE
(1)                                  ;       DEVICE"CSR"REGISTER
(1)                                                                   ;SET:
(1)  006752  004737  015420          TST37: JSR    PC,     @#INITD    ;BIT2 = MAINTENANCE
(1)                                                                   ;BIT3 = CLEAR MOS
(1)                                                                   ;BIT8 = MASTER XMTR SCAN ENB
(1)                                  ;WAIT FOR MOS TO CLEAR
(1)                                                                   ;SET:
(1)                                  ;                                ;BIT0 = RECEIVER ENABLE
(1)                                  ;
(1)  006756  052777  000040  172340        BIS    #BIT5,  @TCR       ;SET XMTR CONTROL BIT, LINE 5
(1)  006764  017701  172330          LOP37: MOV    @CSR,   R1         ;WAIT FOR XMTR READY
(1)  006770  100375                        BPL    LOP37
(1)  006772  012777  000377  172326        MOV    #377,   @TBUF      ;SEND A RUBOUT
(1)  007000  005000                        CLR    R0                 ;CLEAR COUNTER
(1)  007002  105305                  1$:   DECB   R5                 ;SHORT WAIT LOOP
(1)  007004  001376                        BNE    1$
(1)  007006  017701  172310                MOV    @RBUF,R1           ;WAIT FOR CHAR. PRES.
(1)  007012  100405                        BMI    3$                 ;BRANCH WHEN FOUND
(1)  007014  005200                        INC    R0                 ;TIME COUNTER
(1)  007016  001371                        BNE    1$                 ;BRANCH IF NOT TIME-OUT
(1)  007020  017702  172274                MOV    @CSR,   R2         ;SAVE CSR
(1)  007024  104002                        HLT+2                     ;CHARACTER READY DIDN'T SET
(1)                                                                   ;R1 = CONTENTS OF RBUF
(1)                                                                   ;R2 = CONTENTS OF CSR
(1)  007026  050037  001346          3$:   BIS    R0,TIMER           ;SAVE THE TIMER
(1)  007032  032701  070000                BIT    #70000, R1         ;CHECK FOR ERROR BITS
```

```
(1)  007036  001401                        BEQ     .+4             ;BRANCH IF NONE
(1)  007040  104001                        HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)                                                                ;BIT14=UART OVERRUN
(1)                                                                ;BIT13=FRAMING ERROR
(1)                                                                ;BIT12=PARITY ERROR
(1)  007042  010102                 4$:    MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  007044  042702  170377                BIC     #170377,R2      ;MASK LINE#
(1)  007050  000302                        SWAB    R2              ;LINE # IN LOW BYTE
(1)  007052  122702  000005                CMPB    #5.,    R2      ;CHECK LINE #
(1)  007056  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007060  104001                        HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)                                                                ;BITS8-11 = LINE #
(1)  007062  117702  172276         5$:    MOVB    @DJLEN, R2      ;GET MASK OF CHARRACTER
(1)  007066  130201                        BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  007070  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007072  104002                        HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                                ;R1=DATA FROM FI/FO
(1)                                                                ;R2=MASK (BITS SET NOT EXPECTED)
(1)  007074  105102                 6$:    COMB    R2              ;REVERSE THE MASK
(1)  007076  120102                        CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  007100  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007102  104002                        HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                ;R2=DATA (LOW BYTE) EXPECTED
(1)  007104  017701  172212         7$:    MOV     @RBUF,  R1      ;READ FI/FO
(1)  007110  100001                        BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  007112  104001                        HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)  007114  017701  172200         8$:    MOV     @CSR,   R1      ;SAVE THE CSR
(1)  007120  022701  100405                CMP     #100405,R1      ;CHECK THE CSR
(1)  007124  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007126  104001                        HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  007130  005077  172170                CLR     @TCR            ;CLEAR TCR
(1)  007134  005077  172160                CLR     @CSR            ;CLEAR CSR
(1)  007140  104400                        SCOPE
4597
(1)                                ;*************************************************************************
(1)                                ;TEST 40:       TEST THAT LINE 6 CAN TRANSMIT AND
(1)                                ;               RECEIVE A CHARACTER. (377)
(1)                                ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                ;       4$:  CHECKS THAT RIGHT LINE # (6) IN FI/FO
(1)                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                ;       8$:  CHECKS THAT DONE CLEARS
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                ;*************************************************************************
(1)
(1)                                ;               INITIALIZE
(1)                                ;       DEVICE"CSR"REGISTER
(1)                                                             SET:
(1)  007142  004737  015420        TST40:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
```

```
(1)                                                                 ;BIT3 = CLEAR MOS
(1)                                                                 ;BIT8 = MASTER XMTR SCAN ENB
(1)                                 ;WAIT FOR MOS TO CLEAR          SET:
(1)                                 ;                               ;BIT0 = RECEIVER ENABLE
(1)                                 ;
(1)                                 ;
(1) 007146  052777  000100  172150          BIS     #BIT6,  @TCR    ;SET XMTR CONTROL BIT, LINE 6
(1) 007154  017701  172140          LOP40:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1) 007160  100375                          BPL     LOP40
(1) 007162  012777  000377  172136          MOV     #377,   @TBUF   ;SEND A RUBOUT
(1) 007170  005000                          CLR     R0              ;CLEAR COUNTER
(1) 007172  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
(1) 007174  001376                          BNE     1$
(1) 007176  017701  172120                  MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1) 007202  100405                          BMI     3$              ;BRANCH WHEN FOUND
(1) 007204  005200                          INC     R0              ;TIME COUNTER
(1) 007206  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
(1) 007210  017702  172104                  MOV     @CSR,   R2      ;SAVE CSR
(1) 007214  104002                          HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)                                                                 ;R2 = CONTENTS OF CSR
(1) 007216  050037  001346          3$:     BIS     R0,TIMER        ;SAVE THE TIMER
(1) 007222  032701  070000                  BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1) 007226  001401                          BEQ     .+4             ;BRANCH IF NONE
(1) 007230  104001                          HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)                                                                 ;BIT14=UART OVERRUN
(1)                                                                 ;BIT13=FRAMING ERROR
(1)                                                                 ;BIT12=PARITY ERROR
(1) 007232  010102                  4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
(1) 007234  042702  170377                  BIC     #170377,R2      ;MASK LINE#
(1) 007240  000302                          SWAB    R2              ;LINE # IN LOW BYTE
(1) 007242  122702  000006                  CMPB    #6.,    R2      ;CHECK LINE #
(1) 007246  001401                          BEQ     .+4             ;BRANCH IF OK
(1) 007250  104001                          HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)                                                                 ;BITS8-11 = LINE #
(1) 007252  117702  172106          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1) 007256  130201                          BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1) 007260  001401                          BEQ     .+4             ;BRANCH IF OK
(1) 007262  104002                          HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                                 ;R1=DATA FROM FI/FO
(1)                                                                 ;R2=MASK (BITS SET NOT EXPECTED)
(1) 007264  105102                  6$:     COMB    R2              ;REVERSE THE MASK
(1) 007266  120102                          CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1) 007270  001401                          BEQ     .+4             ;BRANCH IF OK
(1) 007272  104002                          HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                 ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                 ;R2=DATA (LOW BYTE) EXPECTED
(1) 007274  017701  172022          7$:     MOV     @RBUF,  R1      ;READ FI/FO
(1) 007300  100001                          BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1) 007302  104001                          HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1) 007304  017701  172010          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
(1) 007310  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
(1) 007314  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
(1)  007316  104001                        HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                  ;R1 = CONTENTS OF CSR
(1)  007320  005077  172000                 CLR     @TCR            ;CLEAR TCR
(1)  007324  005077  171770                 CLR     @CSR            ;CLEAR CSR
(1)  007330  104400                         SCOPE
 4598
(1)                              ;**************************************************************************
(1)                              ;TEST 41:       TEST THAT LINE 7 CAN TRANSMIT AND
(1)                              ;               RECEIVE A CHARACTER. (377)
(1)                              ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                              ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                              ;       4$:  CHECKS THAT RIGHT LINE # (7) IN FI/FO
(1)                              ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                              ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                              ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                              ;       8$:  CHECKS THAT DONE CLEARS
(1)                              ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                              ;**************************************************************************
(1)
(1)                              ;               INITIALIZE
(1)                              ;       DEVICE"CSR"REGISTER
(1)                                                                  ;SET:
(1)  007332  004737  015420      TST41: JSR     PC,     @#INITD     ;BIT2 = MAINTENANCE
(1)                                                                  ;BIT3 = CLEAR MOS
(1)                                                                  ;BIT8 = MASTER XMTR SCAN ENB
(1)                              ;WAIT FOR MOS TO CLEAR
(1)                              ;
(1)                              ;
(1)                              ;
(1)                                                                  ;SET:
(1)                                                                  ;BIT0 = RECEIVER ENABLE
(1)  007336  052777  000200  171760          BIS     #BIT7,  @TCR    ;SET XMTR CONTROL BIT, LINE 7
(1)  007344  017701  171750          LOP41:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  007350  100375                          BPL     LOP41
(1)  007352  012777  000377  171746          MOV     #377,   @TBUF   ;SEND A RUBOUT
(1)  007360  005000                          CLR     R0              ;CLEAR COUNTER
(1)  007362  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
(1)  007364  001376                          BNE     1$
(1)  007366  017701  171730                  MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  007372  100405                          BMI     3$              ;BRANCH WHEN FOUND
(1)  007374  005200                          INC     R0              ;TIME COUNTER
(1)  007376  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
(1)  007400  017702  171714                  MOV     @CSR,   R2      ;SAVE CSR
(1)  007404  104002                          HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                                  ;R1 = CONTENTS OF RBUF
(1)                                                                  ;R2 = CONTENTS OF CSR
(1)  007406  050037  001346          3$:     BIS     R0,TIMER        ;SAVE THE TIMER
(1)  007412  032701  070000                  BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1)  007416  001401                          BEQ     .+4             ;BRANCH IF NONE
(1)  007420  104001                          HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                                  ;R1 = CONTENTS OF RBUF
(1)                                                                  ;BIT14=UART OVERRUN
(1)                                                                  ;BIT13=FRAMING ERROR
(1)                                                                  ;BIT12=PARITY ERROR
(1)  007422  010102                  4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  007424  042702  170377                  BIC     #170377,R2      ;MASK LINE#
(1)  007430  000302                          SWAB    R2              ;LINE # IN LOW BYTE
(1)  007432  122702  000007                  CMPB    #7.,    R2      ;CHECK LINE #
```

```
(1)  007436  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007440  104001                        HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)                                                                ;BITS8-11 = LINE #
(1)  007442  117702  171716        5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  007446  130201                        BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  007450  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007452  104002                        HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                                ;R1=DATA FROM FI/FO
(1)                                                                ;R2=MASK (BITS SET NOT EXPECTED)
(1)  007454  105102        6$:     COMB    R2                      ;REVERSE THE MASK
(1)  007456  120102                        CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  007460  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007462  104002                        HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                ;R2=DATA (LOW BYTE) EXPECTED
(1)  007464  017701  171632        7$:     MOV     @RBUF,  R1      ;READ FI/FO
(1)  007470  100001                        BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  007472  104001                        HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)  007474  017701  171620        8$:     MOV     @CSR,   R1      ;SAVE THE CSR
(1)  007500  022701  100405                CMP     #100405,R1      ;CHECK THE CSR
(1)  007504  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  007506  104001                        HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                ;R1 = CONTENTS OF CSR
(1)  007510  005077  171610                CLR     @TCR            ;CLEAR TCR
(1)  007514  005077  171600                CLR     @CSR            ;CLEAR CSR
(1)  007520  104400                        SCOPE
4599 007522  005237  001364                INC     DJLEN
4600 007526  012737  007534  016070        MOV     #.+6,   LAD
4601
(1)                                ;*****************************************************************************
(1)                                ;TEST 42:       TEST THAT LINE 8 CAN TRANSMIT AND
(1)                                ;               RECEIVE A CHARACTER. (377)
(1)                                ;
(1)                                ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                ;       4$:  CHECKS THAT RIGHT LINE # (8) IN FI/FO
(1)                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                ;       8$:  CHECKS THAT DONE CLEARS
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                ;*****************************************************************************
(1)
(1)                                ;               INITIALIZE
(1)                                ;       DEVICE"CSR"REGISTER
(1)                                ;                                               SET:
(1)  007534  004737  015420        TST42:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                                ;BIT3 = CLEAR MOS
(1)                                                                ;BIT8 = MASTER XMTR SCAN ENB
(1)                                ;WAIT FOR MOS TO CLEAR
(1)                                ;                               SET:
(1)                                ;                               ;BIT0 = RECEIVER ENABLE
(1)                                ;
(1)  007540  052777  000400  171556        BIS     #BIT8,  @TCR    ;SET XMTR CONTROL BIT, LINE 8
(1)  007546  017701  171546        LOP42:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
```

```
(1)  007552  100375                    BPL     LOP42
(1)  007554  012777  000377  171544    MOV     #377,   @TBUF   ;SEND A RUBOUT
(1)  007562  005000                    CLR     R0              ;CLEAR COUNTER
(1)  007564  105305              1$:   DECB    R5              ;SHORT WAIT LOOP
(1)  007566  001376                    BNE     1$
(1)  007570  017701  171526            MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  007574  100405                    BMI     3$              ;BRANCH WHEN FOUND
(1)  007576  005200                    INC     R0              ;TIME COUNTER
(1)  007600  001371                    BNE     1$              ;BRANCH IF NOT TIME-OUT
(1)  007602  017702  171512            MOV     @CSR,   R2      ;SAVE CSR
(1)  007606  104002                    HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)                                                            ;R2 = CONTENTS OF CSR
(1)  007610  050037  001346      3$:   BIS     R0,TIMER        ;SAVE THE TIMER
(1)  007614  032701  070000            BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1)  007620  001401                    BEQ     .+4             ;BRANCH IF NONE
(1)  007622  104001                    HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)                                                            ;BIT14=UART OVERRUN
(1)                                                            ;BIT13=FRAMING ERROR
(1)                                                            ;BIT12=PARITY ERROR
(1)  007624  010102              4$:   MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  007626  042702  170377            BIC     #170377,R2      ;MASK LINE#
(1)  007632  000302                    SWAB    R2              ;LINE # IN LOW BYTE
(1)  007634  122702  000010            CMPB    #8.,    R2      ;CHECK LINE #
(1)  007640  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  007642  104001                    HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)                                                            ;BITS8-11 = LINE #
(1)  007644  117702  171514      5$:   MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  007650  130201                    BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  007652  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  007654  104002                    HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                            ;R1=DATA FROM FI/FO
(1)                                                            ;R2=MASK (BITS SET NOT EXPECTED)
(1)  007656  105102              6$:   COMB    R2              ;REVERSE THE MASK
(1)  007660  120102                    CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  007662  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  007664  104002                    HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                            ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                            ;R2=DATA (LOW BYTE) EXPECTED
(1)  007666  017701  171430      7$:   MOV     @RBUF,  R1      ;READ FI/FO
(1)  007672  100001                    BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  007674  104001                    HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)  007676  017701  171416      8$:   MOV     @CSR,   R1      ;SAVE THE CSR
(1)  007702  022701  100405            CMP     #100405,R1      ;CHECK THE CSR
(1)  007706  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  007710  104001                    HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                            ;R1 = CONTENTS OF CSR
(1)  007712  005077  171406            CLR     @TCR            ;CLEAR TCR
(1)  007716  005077  171376            CLR     @CSR            ;CLEAR CSR
(1)  007722  104400                    SCOPE
4602
(1)
(1)                              ;*****************************************************************************
(1)                              ;TEST 43:       TEST THAT LINE 9 CAN TRANSMIT AND
```

```
(1)                                     ;               RECEIVE A CHARACTER. (377)
(1)                                     ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                     ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                     ;       4$:  CHECKS THAT RIGHT LINE # (9) IN FI/FO
(1)                                     ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                     ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                     ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                     ;       8$:  CHECKS THAT DONE CLEARS
(1)                                     ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                     ;*********************************************************************
(1)
(1)                                     ;               INITIALIZE
(1)                                     ;       DEVICE"CSR"REGISTER
(1)                                     ;                                                       SET:
(1)  007724  004737  015420             TST43:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                                     ;BIT3 = CLEAR MOS
(1)                                                                     ;BIT8 = MASTER XMTR SCAN ENB
(1)                                     ;WAIT FOR MOS TO CLEAR
(1)                                     ;                                               SET:
(1)                                     ;                               ;BIT0 = RECEIVER ENABLE
(1)                                     ;
(1)  007730  052777  001000  171366             BIS     #BIT9,  @TCR    ;SET XMTR CONTROL BIT, LINE 9
(1)  007736  017701  171356             LOP43:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  007742  100375                             BPL     LOP43
(1)  007744  012777  000377  171354             MOV     #377,   @TBUF   ;SEND A RUBOUT
(1)  007752  005000                             CLR     R0              ;CLEAR COUNTER
(1)  007754  105305                     1$:     DECB    R5              ;SHORT WAIT LOOP
(1)  007756  001376                             BNE     1$
(1)  007760  017701  171336                     MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  007764  100405                             BMI     3$              ;BRANCH WHEN FOUND
(1)  007766  005200                             INC     R0              ;TIME COUNTER
(1)  007770  001371                             BNE     1$              ;BRANCH IF NOT TIME-OUT
(1)  007772  017702  171322                     MOV     @CSR,   R2      ;SAVE CSR
(1)  007776  104002                             HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;R2 = CONTENTS OF CSR
(1)  010000  050037  001346             3$:     BIS     R0,TIMER        ;SAVE THE TIMER
(1)  010004  032701  070000                     BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1)  010010  001401                             BEQ     .+4             ;BRANCH IF NONE
(1)  010012  104001                             HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BIT14=UART OVERRUN
(1)                                                                     ;BIT13=FRAMING ERROR
(1)                                                                     ;BIT12=PARITY ERROR
(1)  010014  010102                     4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  010016  042702  170377                     BIC     #170377,R2      ;MASK LINE#
(1)  010022  000302                             SWAB    R2              ;LINE # IN LOW BYTE
(1)  010024  122702  000011                     CMPB    #9.,    R2      ;CHECK LINE #
(1)  010030  001401                             BEQ     .+4             ;BRANCH IF OK
(1)  010032  104001                             HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BITS8-11 = LINE #
(1)  010034  117702  171324             5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  010040  130201                             BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  010042  001401                             BEQ     .+4             ;BRANCH IF OK
(1)  010044  104002                             HLT+2                   ;WRONG CHARACTER LENGTH
```

```
(1)
(1)
(1)  010046  105102                6$:     COMB    R2                ;R1=DATA FROM FI/FO
(1)  010050  120102                        CMPB    R1,     R2        ;R2=MASK (BITS SET NOT EXPECTED)
(1)  010052  001401                        BEQ     .+4               ;REVERSE THE MASK
(1)  010054  104002                        HLT+2                     ;CHECK THE ACTURAL DATA
(1)                                                                   ;BRANCH IF OK
(1)                                                                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                   ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                   ;R2=DATA (LOW BYTE) EXPECTED
(1)  010056  017701  171240         7$:     MOV     @RBUF,  R1        ;READ FI/FO
(1)  010062  100001                        BPL     .+4               ;BRANCH IF CHAR PRESENT NOT SET
(1)  010064  104001                        HLT+1                     ;CHARACTER PRESENT STAYED SET
(1)                                                                   ;R1 = CONTENTS OF RBUF
(1)  010066  017701  171226         8$:     MOV     @CSR,   R1        ;SAVE THE CSR
(1)  010072  022701  100405                 CMP     #100405,R1        ;CHECK THE CSR
(1)  010076  001401                        BEQ     .+4               ;BRANCH IF OK
(1)  010100  104001                        HLT+1                     ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                   ;R1 = CONTENTS OF CSR
(1)  010102  005077  171216                 CLR     @TCR              ;CLEAR TCR
(1)  010106  005077  171206                 CLR     @CSR              ;CLEAR CSR
(1)  010112  104400                        SCOPE
4603
(1)                                 ;*************************************************************************
(1)                                 ;TEST 44:       TEST THAT LINE 10 CAN TRANSMIT AND
(1)                                 ;               RECEIVE A CHARACTER. (377)
(1)                                 ;
(1)                                 ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                 ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                 ;       4$:  CHECKS THAT RIGHT LINE # (10) IN FI/FO
(1)                                 ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                 ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                 ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                 ;       8$:  CHECKS THAT DONE CLEARS
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                 ;*************************************************************************
(1)
(1)                                 ;               INITIALIZE
(1)                                 ;       DEVICE"CSR"REGISTER
(1)                                 ;                                                       SET:
(1)  010114  004737  015420         TST44:  JSR     PC,     @#INITD   ;BIT2 = MAINTENANCE
(1)                                                                   ;BIT3 = CLEAR MOS
(1)                                                                   ;BIT8 = MASTER XMTR SCAN ENB
(1)                                 ;WAIT FOR MOS TO CLEAR
(1)                                 ;                                                       SET:
(1)                                 ;                                                       ;BIT0 = RECEIVER ENABLE
(1)                                 ;
(1)  010120  052777  002000  171176         BIS     #BIT10, @TCR      ;SET XMTR CONTROL BIT, LINE 10
(1)  010126  017701  171166         LOP44:  MOV     @CSR,   R1        ;WAIT FOR XMTR READY
(1)  010132  100375                         BPL     LOP44
(1)  010134  012777  000377  171164         MOV     #377,   @TBUF     ;SEND A RUBOUT
(1)  010142  005000                         CLR     R0                ;CLEAR COUNTER
(1)  010144  105305                 1$:     DECB    R5                ;SHORT WAIT LOOP
(1)  010146  001376                         BNE     1$
(1)  010150  017701  171146                 MOV     @RBUF,R1          ;WAIT FOR CHAR. PRES.
(1)  010154  100405                         BMI     3$                ;BRANCH WHEN FOUND
(1)  010156  005200                         INC     R0                ;TIME COUNTER
(1)  010160  001371                         BNE     1$                ;BRANCH IF NOT TIME-OUT
(1)  010162  017702  171132                 MOV     @CSR,   R2        ;SAVE CSR
```

```
(1)  010166  104002                          HLT+2                  ;CHARACTER READY DIDN'T SET
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)                                                                 ;R2 = CONTENTS OF CSR
(1)  010170  050037  001346        3$:       BIS     R0,TIMER       ;SAVE THE TIMER
(1)  010174  032701  070000                   BIT     #70000, R1     ;CHECK FOR ERROR BITS
(1)  010200  001401                           BEQ     .+4            ;BRANCH IF NONE
(1)  010202  104001                           HLT+1                  ;ERROR IN RECEIVED CHAR
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)                                                                 ;BIT14=UART OVERRUN
(1)                                                                 ;BIT13=FRAMING ERROR
(1)                                                                 ;BIT12=PARITY ERROR
(1)  010204  010102        4$:       MOV     R1,     R2             ;DUPLICATE DATA WORD
(1)  010206  042702  170377           BIC     #170377,R2             ;MASK LINE#
(1)  010212  000302                   SWAB    R2                     ;LINE # IN LOW BYTE
(1)  010214  122702  000012           CMPB    #10.,   R2             ;CHECK LINE #
(1)  010220  001401                   BEQ     .+4                    ;BRANCH IF OK
(1)  010222  104001                   HLT+1                          ;WRONG LINE # RECEIVED
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)                                                                 ;BITS8-11 = LINE #
(1)  010224  117702  171134        5$:       MOVB    @DJLEN, R2     ;GET MASK OF CHARACTER
(1)  010230  130201                   BITB    R2,     R1             ;CHECK CHAR LENGTH.
(1)  010232  001401                   BEQ     .+4                    ;BRANCH IF OK
(1)  010234  104002                   HLT+2                          ;WRONG CHARACTER LENGTH
(1)                                                                 ;R1=DATA FROM FI/FO
(1)                                                                 ;R2=MASK (BITS SET NOT EXPECTED)
(1)  010236  105102        6$:       COMB    R2                     ;REVERSE THE MASK
(1)  010240  120102                   CMPB    R1,     R2             ;CHECK THE ACTURAL DATA
(1)  010242  001401                   BEQ     .+4                    ;BRANCH IF OK
(1)  010244  104002                   HLT+2                          ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                 ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                 ;R2=DATA (LOW BYTE) EXPECTED
(1)  010246  017701  171050        7$:       MOV     @RBUF,  R1     ;READ FI/FO
(1)  010252  100001                   BPL     .+4                    ;BRANCH IF CHAR PRESENT NOT SET
(1)  010254  104001                   HLT+1                          ;CHARACTER PRESENT STAYED SET
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)  010256  017701  171036        8$:       MOV     @CSR,   R1     ;SAVE THE CSR
(1)  010262  022701  100405           CMP     #100405,R1             ;CHECK THE CSR
(1)  010266  001401                   BEQ     .+4                    ;BRANCH IF OK
(1)  010270  104001                   HLT+1                          ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                 ;R1 = CONTENTS OF CSR
(1)  010272  005077  171026           CLR     @TCR                   ;CLEAR TCR
(1)  010276  005077  171016           CLR     @CSR                   ;CLEAR CSR
(1)  010302  104400                   SCOPE
4604
(1)                                 ;******************************************************************************
(1)                                 ;TEST 45:       TEST THAT LINE 11 CAN TRANSMIT AND
(1)                                 ;               RECEIVE A CHARACTER. (377)
(1)                                 ;
(1)                                 ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                 ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                 ;       4$:  CHECKS THAT RIGHT LINE # (11) IN FI/FO
(1)                                 ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                 ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                 ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                 ;       8$:  CHECKS THAT DONE CLEARS
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                 ;******************************************************************************
```

```
(1)                                     ;               INITIALIZE
(1)                                     ;       DEVICE"CSR"REGISTER
(1)                                     ;                                       SET:
(1)  010304  004737  015420             TST45:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                                     ;BIT3 = CLEAR MOS
(1)                                                                     ;BIT8 = MASTER XMTR SCAN ENB
(1)                                     ;WAIT FOR MOS TO CLEAR
(1)                                     ;                               SET:
(1)                                     ;                               ;BIT0 = RECEIVER ENABLE
(1)                                     ;
(1)  010310  052777  004000  171006             BIS     #BIT11, @TCR    ;SET XMTR CONTROL BIT, LINE 11
(1)  010316  017701  170776             LOP45:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  010322  100375                             BPL     LOP45
(1)  010324  012777  000377  170774             MOV     #377,   @TBUF   ;SEND A RUBOUT
(1)  010332  005000                             CLR     R0              ;CLEAR COUNTER
(1)  010334  105305                     1$:     DECB    R5              ;SHORT WAIT LOOP
(1)  010336  001376                             BNE     1$
(1)  010340  017701  170756                     MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  010344  100405                             BMI     3$              ;BRANCH WHEN FOUND
(1)  010346  005200                             INC     R0              ;TIME COUNTER
(1)  010350  001371                             BNE     1$              ;BRANCH IF NOT TIME-OUT
(1)  010352  017702  170742                     MOV     @CSR,   R2      ;SAVE CSR
(1)  010356  104002                             HLT+2                   ;CHARACTER READY DIDN'T SET
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;R2 = CONTENTS OF CSR
(1)  010360  050037  001346             3$:     BIS     R0,TIMER        ;SAVE THE TIMER
(1)  010364  032701  070000                     BIT     #70000, R1      ;CHECK FOR ERROR BITS
(1)  010370  001401                             BEQ     .+4             ;BRANCH IF NONE
(1)  010372  104001                             HLT+1                   ;ERROR IN RECEIVED CHAR
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BIT14=UART OVERRUN
(1)                                                                     ;BIT13=FRAMING ERROR
(1)                                                                     ;BIT12=PARITY ERROR
(1)  010374  010102                     4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  010376  042702  170377                     BIC     #170377,R2      ;MASK LINE#
(1)  010402  000302                             SWAB    R2              ;LINE # IN LOW BYTE
(1)  010404  122702  000013                     CMPB    #11.,   R2      ;CHECK LINE #
(1)  010410  001401                             BEQ     .+4             ;BRANCH IF OK
(1)  010412  104001                             HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BITS8-11 = LINE #
(1)  010414  117702  170744             5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  010420  130201                             BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  010422  001401                             BEQ     .+4             ;BRANCH IF OK
(1)  010424  104002                             HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                                     ;R1=DATA FROM FI/FO
(1)                                                                     ;R2=MASK (BITS SET NOT EXPECTED)
(1)  010426  105102                     6$:     COMB    R2              ;REVERSE THE MASK
(1)  010430  120102                             CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  010432  001401                             BEQ     .+4             ;BRANCH IF OK
(1)  010434  104002                             HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                     ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                     ;R2=DATA (LOW BYTE) EXPECTED
(1)  010436  017701  170660             7$:     MOV     @RBUF,  R1      ;READ FI/FO
(1)  010442  100001                             BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
```

```
(1)  010444  104001                          HLT+1                    ;CHARACTER PRESENT STAYED SET
(1)                                                                    ;R1 = CONTENTS OF RBUF
(1)  010446  017701  170646          8$:     MOV     @CSR,   R1       ;SAVE THE CSR
(1)  010452  022701  100405                  CMP     #100405,R1       ;CHECK THE CSR
(1)  010456  001401                          BEQ     .+4              ;BRANCH IF OK
(1)  010460  104001                          HLT+1                    ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                    ;R1 = CONTENTS OF CSR
(1)  010462  005077  170636                  CLR     @TCR             ;CLEAR TCR
(1)  010466  005077  170626                  CLR     @CSR             ;CLEAR CSR
(1)  010472  104400                          SCOPE
4605 010474  005237  001364                  INC     DJLEN
4606 010500  012737  010506  016070          MOV     #.+6,   LAD
4607
(1)                                ;*************************************************************************
(1)                                ;TEST 46:       TEST THAT LINE 12 CAN TRANSMIT AND
(1)                                ;               RECEIVE A CHARACTER. (377)
(1)                                ;
(1)                                ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                ;       4$:  CHECKS THAT RIGHT LINE # (12) IN FI/FO
(1)                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                ;       8$:  CHECKS THAT DONE CLEARS
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                ;*************************************************************************
(1)
(1)                                ;               INITIALIZE
(1)                                ;       DEVICE"CSR"REGISTER
(1)                                ;                                                SET:
(1)  010506  004737  015420        TST46:  JSR     PC,     @#INITD  ;BIT2 = MAINTENANCE
(1)                                                                 ;BIT3 = CLEAR MOS
(1)                                                                 ;BIT8 = MASTER XMTR SCAN ENB
(1)                                ;WAIT FOR MOS TO CLEAR
(1)                                ;                                                SET:
(1)                                ;                                                ;BIT0 = RECEIVER ENABLE
(1)                                ;
(1)  010512  052777  010000  170604          BIS     #BIT12, @TCR     ;SET XMTR CONTROL BIT, LINE 12
(1)  010520  017701  170574          LOP46:  MOV     @CSR,   R1       ;WAIT FOR XMTR READY
(1)  010524  100375                          BPL     LOP46
(1)  010526  012777  000377  170572          MOV     #377,   @TBUF    ;SEND A RUBOUT
(1)  010534  005000                          CLR     R0               ;CLEAR COUNTER
(1)  010536  105305                  1$:     DECB    R5               ;SHORT WAIT LOOP
(1)  010540  001376                          BNE     1$
(1)  010542  017701  170554                  MOV     @RBUF,R1         ;WAIT FOR CHAR. PRES.
(1)  010546  100405                          BMI     3$               ;BRANCH WHEN FOUND
(1)  010550  005200                          INC     R0               ;TIME COUNTER
(1)  010552  001371                          BNE     1$               ;BRANCH IF NOT TIME-OUT
(1)  010554  017702  170540                  MOV     @CSR,   R2       ;SAVE CSR
(1)  010560  104002                          HLT+2                    ;CHARACTER READY DIDN'T SET
(1)                                                                    ;R1 = CONTENTS OF RBUF
(1)                                                                    ;R2 = CONTENTS OF CSR
(1)  010562  050037  001346          3$:     BIS     R0,TIMER         ;SAVE THE TIMER
(1)  010566  032701  070000                  BIT     #70000, R1       ;CHECK FOR ERROR BITS
(1)  010572  001401                          BEQ     .+4              ;BRANCH IF NONE
(1)  010574  104001                          HLT+1                    ;ERROR IN RECEIVED CHAR
(1)                                                                    ;R1 = CONTENTS OF RBUF
```

```
(1)                                                     ;BIT14=UART OVERRUN
(1)                                                     ;BIT13=FRAMING ERROR
(1)                                                     ;BIT12=PARITY ERROR
(1)  010576  010102                 4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
(1)  010600  042702  170377                 BIC     #170377,R2      ;MASK LINE#
(1)  010604  000302                         SWAB    R2              ;LINE # IN LOW BYTE
(1)  010606  122702  000014                 CMPB    #12.,   R2      ;CHECK LINE #
(1)  010612  001401                         BEQ     .+4             ;BRANCH IF OK
(1)  010614  104001                         HLT+1                   ;WRONG LINE # RECEIVED
(1)                                                     ;R1 = CONTENTS OF RBUF
(1)                                                     ;BITS8-11 = LINE #
(1)  010616  117702  170542         5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  010622  130201                         BITB    R2,     R1      ;CHECK CHAR LENGTH.
(1)  010624  001401                         BEQ     .+4             ;BRANCH IF OK
(1)  010626  104002                         HLT+2                   ;WRONG CHARACTER LENGTH
(1)                                                     ;R1=DATA FROM FI/FO
(1)                                                     ;R2=MASK (BITS SET NOT EXPECTED)
(1)  010630  105102                 6$:     COMB    R2              ;REVERSE THE MASK
(1)  010632  120102                         CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
(1)  010634  001401                         BEQ     .+4             ;BRANCH IF OK
(1)  010636  104002                         HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
(1)                                                     ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                     ;R2=DATA (LOW BYTE) EXPECTED
(1)  010640  017701  170456         7$:     MOV     @RBUF,  R1      ;READ FI/FO
(1)  010644  100001                         BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
(1)  010646  104001                         HLT+1                   ;CHARACTER PRESENT STAYED SET
(1)                                                     ;R1 = CONTENTS OF RBUF
(1)  010650  017701  170444         8$:     MOV     @CSR,   R1      ;SAVE THE CSR
(1)  010654  022701  100405                 CMP     #100405,R1      ;CHECK THE CSR
(1)  010660  001401                         BEQ     .+4             ;BRANCH IF OK
(1)  010662  104001                         HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                     ;R1 = CONTENTS OF CSR
(1)  010664  005077  170434                 CLR     @TCR            ;CLEAR TCR
(1)  010670  005077  170424                 CLR     @CSR            ;CLEAR CSR
(1)  010674  104400                         SCOPE
4608
(1)                         ;*****************************************************************************
(1)                         ;TEST 47:       TEST THAT LINE 13 CAN TRANSMIT AND
(1)                         ;               RECEIVE A CHARACTER. (377)
(1)                         ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                         ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                         ;       4$:  CHECKS THAT RIGHT LINE # (13) IN FI/FO
(1)                         ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                         ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                         ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                         ;       8$:  CHECKS THAT DONE CLEARS
(1)                         ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DD3 SERIES
(1)                         ;*****************************************************************************
(1)
(1)                         ;               INITIALIZE
(1)                         ;       DEVICE"CSR"REGISTER
(1)                                                             SET:
(1)  010676  004737  015420 TST47:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                             ;BIT3 = CLEAR MOS
(1)                                                             ;BIT8 = MASTER XMTR SCAN ENB
(1)
(1)                         ;WAIT FOR MOS TO CLEAR
```

```
(1)                                         ;                                   SET:
(1)                                         ;                                   ;BIT0 = RECEIVER ENABLE
(1)                                         ;
(1)  010702  052777  020000  170414         BIS     #BIT13, @TCR        ;SET XMTR CONTROL BIT, LINE 13
(1)  010710  017701  170404         LOP47:  MOV     @CSR,   R1          ;WAIT FOR XMTR READY
(1)  010714  100375                         BPL     LOP47
(1)  010716  012777  000377  170402         MOV     #377,   @TBUF       ;SEND A RUBOUT
(1)  010724  005000                         CLR     R0                  ;CLEAR COUNTER
(1)  010726  105305                 1$:     DECB    R5                  ;SHORT WAIT LOOP
(1)  010730  001376                         BNE     1$
(1)  010732  017701  170364                 MOV     @RBUF,R1            ;WAIT FOR CHAR. PRES.
(1)  010736  100405                         BMI     3$                  ;BRANCH WHEN FOUND
(1)  010740  005200                         INC     R0                  ;TIME COUNTER
(1)  010742  001371                         BNE     1$                  ;BRANCH IF NOT TIME-OUT
(1)  010744  017702  170350                 MOV     @CSR,   R2          ;SAVE CSR
(1)  010750  104002                         HLT+2                       ;CHARACTER READY DIDN'T SET
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;R2 = CONTENTS OF CSR
(1)  010752  050037  001346         3$:     BIS     R0,TIMER            ;SAVE THE TIMER
(1)  010756  032701  070000                 BIT     #70000, R1          ;CHECK FOR ERROR BITS
(1)  010762  001401                         BEQ     .+4                 ;BRANCH IF NONE
(1)  010764  104001                         HLT+1                       ;ERROR IN RECEIVED CHAR
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BIT14=UART OVERRUN
(1)                                                                     ;BIT13=FRAMING ERROR
(1)                                                                     ;BIT12=PARITY ERROR
(1)  010766  010102                 4$:     MOV     R1,     R2          ;DUPLICATE DATA WORD
(1)  010770  042702  170377                 BIC     #170377,R2          ;MASK LINE#
(1)  010774  000302                         SWAB    R2                  ;LINE # IN LOW BYTE
(1)  010776  122702  000015                 CMPB    #13.,   R2          ;CHECK LINE #
(1)  011002  001401                         BEQ     .+4                 ;BRANCH IF OK
(1)  011004  104001                         HLT+1                       ;WRONG LINE # RECEIVED
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)                                                                     ;BITS8-11 = LINE #
(1)  011006  117702  170352         5$:     MOVB    @DJLEN, R2          ;GET MASK OF CHARACTER
(1)  011012  130201                         BITB    R2,     R1          ;CHECK CHAR LENGTH.
(1)  011014  001401                         BEQ     .+4                 ;BRANCH IF OK
(1)  011016  104002                         HLT+2                       ;WRONG CHARACTER LENGTH
(1)                                                                     ;R1=DATA FROM FI/FO
(1)                                                                     ;R2=MASK (BITS SET NOT EXPECTED)
(1)  011020  105102                 6$:     COMB    R2                  ;REVERSE THE MASK
(1)  011022  120102                         CMPB    R1,     R2          ;CHECK THE ACTURAL DATA
(1)  011024  001401                         BEQ     .+4                 ;BRANCH IF OK
(1)  011026  104002                         HLT+2                       ;WRONG CHAR LEN OR DATA ERROR
(1)                                                                     ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                                     ;R2=DATA (LOW BYTE) EXPECTED
(1)  011030  017701  170266         7$:     MOV     @RBUF,  R1          ;READ FI/FO
(1)  011034  100001                         BPL     .+4                 ;BRANCH IF CHAR PRESENT NOT SET
(1)  011036  104001                         HLT+1                       ;CHARACTER PRESENT STAYED SET
(1)                                                                     ;R1 = CONTENTS OF RBUF
(1)  011040  017701  170254         8$:     MOV     @CSR,   R1          ;SAVE THE CSR
(1)  011044  022701  100405                 CMP     #100405,R1          ;CHECK THE CSR
(1)  011050  001401                         BEQ     .+4                 ;BRANCH IF OK
(1)  011052  104001                         HLT+1                       ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                                     ;R1 = CONTENTS OF CSR
(1)  011054  005077  170244                 CLR     @TCR                ;CLEAR TCR
```

```
  (1)  011060  005077  170234                CLR     @CSR            ;CLEAR CSR
  (1)  011064  104400                        SCOPE
 4609
  (1)                              ;*************************************************************************
  (1)                              ;TEST 50:       TEST THAT LINE 14 CAN TRANSMIT AND
  (1)                              ;               RECEIVE A CHARACTER. (377)
  (1)                              ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
  (1)                              ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
  (1)                              ;       4$:  CHECKS THAT RIGHT LINE # (14) IN FI/FO
  (1)                              ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
  (1)                              ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
  (1)                              ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
  (1)                              ;       8$:  CHECKS THAT DONE CLEARS
  (1)                              ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
  (1)                              ;*************************************************************************
  (1)
  (1)                              ;               INITIALIZE
  (1)                              ;       DEVICE"CSR"REGISTER
  (1)                                                              SET:
  (1)  011066  004737  015420      TST50:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
  (1)                                                              ;BIT3 = CLEAR MOS
  (1)                                                              ;BIT8 = MASTER XMTR SCAN ENB,
  (1)                              ;WAIT FOR MOS TO CLEAR
  (1)                                                              SET:
  (1)                              ;                               ;BIT0 = RECEIVER ENABLE
  (1)                              ;
  (1)
  (1)  011072  052777  040000  170224          BIS     #BIT14, @TCR    ;SET XMTR CONTROL BIT, LINE 14
  (1)  011100  017701  170214        LOP50:    MOV     @CSR,   R1      ;WAIT FOR XMTR READY
  (1)  011104  100375                BPL     LOP50
  (1)  011106  012777  000377  170212        MOV     #377,   @TBUF   ;SEND A RUBOUT
  (1)  011114  005000                CLR     R0              ;CLEAR COUNTER
  (1)  011116  105305                1$:     DECB    R5              ;SHORT WAIT LOOP
  (1)  011120  001376                BNE     1$
  (1)  011122  017701  170174        MOV     @RBUF,R1        ;WAIT FOR CHAR. PRES.
  (1)  011126  100405                BMI     3$              ;BRANCH WHEN FOUND
  (1)  011130  005200                INC     R0              ;TIME COUNTER
  (1)  011132  001371                BNE     1$              ;BRANCH IF NOT TIME-OUT
  (1)  011134  017702  170160        MOV     @CSR,   R2      ;SAVE CSR
  (1)  011140  104002                HLT+2                   ;CHARACTER READY DIDN'T SET
  (1)                                                        ;R1 = CONTENTS OF RBUF
  (1)                                                        ;R2 = CONTENTS OF CSR
  (1)  011142  050037  001346        3$:     BIS     R0,TIMER        ;SAVE THE TIMER
  (1)  011146  032701  070000        BIT     #70000, R1      ;CHECK FOR ERROR BITS
  (1)  011152  001401                BEQ     .+4             ;BRANCH IF NONE
  (1)  011154  104001                HLT+1                   ;ERROR IN RECEIVED CHAR
  (1)                                                        ;R1 = CONTENTS OF RBUF
  (1)                                                        ;BIT14=UART OVERRUN
  (1)                                                        ;BIT13=FRAMING ERROR
  (1)                                                        ;BIT12=PARITY ERROR
  (1)  011156  010102                4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
  (1)  011160  042702  170377        BIC     #170377,R2      ;MASK LINE#
  (1)  011164  000302                SWAB    R2              ;LINE # IN LOW BYTE
  (1)  011166  122702  000016        CMPB    #14.,   R2      ;CHECK LINE #
  (1)  011172  001401                BEQ     .+4             ;BRANCH IF OK
  (1)  011174  104001                HLT+1                   ;WRONG LINE # RECEIVED
  (1)                                                        ;R1 = CONTENTS OF RBUF
```

```
(1)
(1)  011176  117702  170162        5$:   MOVB   @DJLEN, R2     ;BITS8-11 = LINE #
(1)  011202  130201                 BITB   R2,     R1          ;GET MASK OF CHARACTER
(1)  011204  001401                 BEQ    .+4                 ;CHECK CHAR LENGTH.
(1)  011206  104002                 HLT+2                      ;BRANCH IF OK
(1)                                                            ;WRONG CHARACTER LENGTH
(1)                                                            ;R1=DATA FROM FI/FO
(1)                                                            ;R2=MASK (BITS SET NOT EXPECTED)
(1)  011210  105102                 6$:   COMB   R2            ;REVERSE THE MASK
(1)  011212  120102                 CMPB   R1,     R2          ;CHECK THE ACTURAL DATA
(1)  011214  001401                 BEQ    .+4                 ;BRANCH IF OK
(1)  011216  104002                 HLT+2                      ;WRONG CHAR LEN OR DATA ERROR
(1)                                                            ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                            ;R2=DATA (LOW BYTE) EXPECTED
(1)  011220  017701  170076         7$:   MOV    @RBUF,  R1    ;READ FI/FO
(1)  011224  100001                 BPL    .+4                 ;BRANCH IF CHAR PRESENT NOT SET
(1)  011226  104001                 HLT+1                      ;CHARACTER PRESENT STAYED SET
(1)                                                            ;R1 = CONTENTS OF RBUF
(1)  011230  017701  170064         8$:   MOV    @CSR,   R1    ;SAVE THE CSR
(1)  011234  022701  100405         CMP    #100405,R1          ;CHECK THE CSR
(1)  011240  001401                 BEQ    .+4                 ;BRANCH IF OK
(1)  011242  104001                 HLT+1                      ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                            ;R1 = CONTENTS OF CSR
(1)  011244  005077  170054         CLR    @TCR                ;CLEAR TCR
(1)  011250  005077  170044         CLR    @CSR                ;CLEAR CSR
(1)  011254  104400                 SCOPE
4610
(1)                                 ;#########################################################################
(1)                                 ;TEST 51:       TEST THAT LINE 15 CAN TRANSMIT AND
(1)                                 ;               RECEIVE A CHARACTER. (377)
(1)                                 ;       1$:  CHECKS THAT CHAR PRESENT IS IN FI/FO IN REASONABLE TIME.
(1)                                 ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
(1)                                 ;       4$:  CHECKS THAT RIGHT LINE # (15) IN FI/FO
(1)                                 ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
(1)                                 ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
(1)                                 ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
(1)                                 ;       8$:  CHECKS THAT DONE CLEARS
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
(1)                                 ;#########################################################################
(1)
(1)                                 ;             INITIALIZE
(1)                                 ;       DEVICE"CSR"REGISTER
(1)                                 ;                                          SET:
(1)  011256  004737  015420         TST51: JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                                ;BIT3 = CLEAR MOS
(1)                                                                ;BIT8 = MASTER XMTR SCAN ENB
(1)                                 ;WAIT FOR MOS TO CLEAR
(1)                                 ;                          SET:
(1)                                 ;                          ;BIT0 = RECEIVER ENABLE
(1)                                 ;
(1)  011262  052777  100000  170034         BIS    #BIT15, @TCR    ;SET XMTR CONTROL BIT, LINE 15
(1)  011270  017701  170024         LOP51: MOV    @CSR,   R1      ;WAIT FOR XMTR READY
(1)  011274  100375                 BPL    LOP51
(1)  011276  012777  000377  170022         MOV    #377,   @TBUF   ;SEND A RUBOUT
(1)  011304  005000                 CLR    R0                     ;CLEAR COUNTER
(1)  011306  105305                 1$:    DECB   R5              ;SHORT WAIT LOOP
(1)  011310  001376                 BNE    1$
```

```
(1)  011312  017701  170004          MOV    @RBUF,R1        ;WAIT FOR CHAR. PRES.
(1)  011316  100405                  BMI    3$              ;BRANCH WHEN FOUND
(1)  011320  005200                  INC    R0              ;TIME COUNTER
(1)  011322  001371                  BNE    1$              ;BRANCH IF NOT TIME-OUT
(1)  011324  017702  167770          MOV    @CSR,  R2       ;SAVE CSR
(1)  011330  104002                  HLT+2                  ;CHARACTER READY DIDN'T SET
(1)                                                         ;R1 = CONTENTS OF RBUF
(1)                                                         ;R2 = CONTENTS OF CSR
(1)  011332  050037  001346  3$:     BIS    R0,TIMER        ;SAVE THE TIMER
(1)  011336  032701  070000          BIT    #70000, R1      ;CHECK FOR ERROR BITS
(1)  011342  001401                  BEQ    .+4             ;BRANCH IF NONE
(1)  011344  104001                  HLT+1                  ;ERROR IN RECEIVED CHAR
(1)                                                         ;R1 = CONTENTS OF RBUF
(1)                                                         ;BIT14=UART OVERRUN
(1)                                                         ;BIT13=FRAMING ERROR
(1)                                                         ;BIT12=PARITY ERROR
(1)  011346  010102  4$:     MOV    R1,    R2              ;DUPLICATE DATA WORD
(1)  011350  042702  170377          BIC    #170377,R2      ;MASK LINE#
(1)  011354  000302                  SWAB   R2              ;LINE # IN LOW BYTE
(1)  011356  122702  000017          CMPB   #15.,  R2       ;CHECK LINE #
(1)  011362  001401                  BEQ    .+4             ;BRANCH IF OK
(1)  011364  104001                  HLT+1                  ;WRONG LINE # RECEIVED
(1)                                                         ;R1 = CONTENTS OF RBUF
(1)                                                         ;BITS8-11 = LINE #
(1)  011366  117702  167772  5$:     MOVB   @DJLEN, R2      ;GET MASK OF CHARACTER
(1)  011372  130201                  BITB   R2,    R1       ;CHECK CHAR LENGTH.
(1)  011374  001401                  BEQ    .+4             ;BRANCH IF OK
(1)  011376  104002                  HLT+2                  ;WRONG CHARACTER LENGTH
(1)                                                         ;R1=DATA FROM FI/FO
(1)                                                         ;R2=MASK (BITS SET NOT EXPECTED)
(1)  011400  105102  6$:     COMB   R2                     ;REVERSE THE MASK
(1)  011402  120102          CMPB   R1,    R2              ;CHECK THE ACTURAL DATA
(1)  011404  001401          BEQ    .+4                    ;BRANCH IF OK
(1)  011406  104002          HLT+2                         ;WRONG CHAR LEN OR DATA ERROR
(1)                                                        ;R1=DATA FROM FI/FO (COMPLETE WORD)
(1)                                                        ;R2=DATA (LOW BYTE) EXPECTED
(1)  011410  017701  167706  7$:     MOV    @RBUF, R1      ;READ FI/FO
(1)  011414  100001          BPL    .+4                    ;BRANCH IF CHAR PRESENT NOT SET
(1)  011416  104001          HLT+1                         ;CHARACTER PRESENT STAYED SET
(1)                                                        ;R1 = CONTENTS OF RBUF
(1)  011420  017701  167674  8$:     MOV    @CSR,  R1      ;SAVE THE CSR
(1)  011424  022701  100405          CMP    #100405,R1     ;CHECK THE CSR
(1)  011430  001401          BEQ    .+4                    ;BRANCH IF OK
(1)  011432  104001          HLT+1                         ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
(1)                                                        ;R1 = CONTENTS OF CSR
(1)  011434  005077  167664          CLR    @TCR           ;CLEAR TCR
(1)  011440  005077  167654          CLR    @CSR           ;CLEAR CSR
(1)  011444  104400          SCOPE
4611 011446  162737  000003  001364  SUB    #3,    DJLEN
4612 011454  005237  001346          INC    TIMER          ;WORSE CASE TIME, ONE CHARACTER
4613 011460  013700  001346          MOV    TIMER, R0      ;DUP TIMER
4614 011464  006200                  ASR    R0             ;/2
4615 011466  006200                  ASR    R0             ;/4
4616 011470  005200                  INC    R0             ;+1
4617 011472  006137  001346          ROL    TIMER          ;TIMER * 2
4618 011476  060037  001346          ADD    R0,    TIMER   ;2.25 TIMES ONE CHARACTER TIME
```

```
 4619  011502  012737  011510  016070          MOV     #.+6,   LAD     ;RESET LOOP ADDRESS
 4620
  (1)                                ;*************************************************************************
  (1)                                ;TEST 52:       TEST THAT CHARACTER PRESENT (BIT15) OF RBUF
  (1)                                ;               IS CLEARED (BY CLEAR MOS).
  (1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-8) E17,E14,E15; M7279; M7280
  (1)                                ;*************************************************************************
  (1)
  (1)                                ;       INITIALIZE
  (1)                                ;
  (1)  011510  004737  015420        TST52:  JSR     PC,@#INITD
  (1)                                ;SET:
  (1)                                ;       BIT2 = MAINTENANCE
  (1)                                ;       BIT3 = CLEAR MOS
  (1)                                ;       BIT8 = TRANS SCAN ENABLE
  (1)                                ;       WAIT FOR BIT4 = MOS CLEAR
  (1)                                ;SET:
  (1)                                ;       BIT0 = RECEIVER ENABLE
  (1)  011514  012777  000001  167602        MOV     #BIT0,  @TCR    ;TRANS CONTROL, LINE0
  (1)  011522  012704  000040                MOV     #40,R4          ;SET UP COUNTER TO OUTPUT 32 CHAR'S
  (1)  011526  017701  167566        1$:     MOV     @CSR,R1         ;WAIT FOR TRANS READY
  (1)  011532  100375                        BPL     1$
  (1)  011534  010477  167566                MOV     R4,@TBUF        ;TRANSMIT COUNT
  (1)  011540  005304                        DEC     R4              ;COUNT DOWN
  (1)  011542  001371                        BNE     1$
  (1)
  (1)  011544  105777  167550        2$:     TSTB    @CSR            ;MAKE SURE DONE IS SET
  (1)  011550  100375                        BPL     2$
  (1)
  (1)  011552  052777  000010  167540        BIS     #BIT3,@CSR      ;CLEAR MOS
  (1)  011560  032777  000020  167532  3$:   BIT     #BIT4,@CSR      ;WAIT FOR CLRMOS TO FINISH
  (1)  011566  001374                        BNE     3$
  (1)
  (1)  011570  017701  167526                MOV     @RBUF,R1        ;CHECK RBUF AND SAVE
  (1)  011574  100001                        BPL     .+4
  (1)  011576  104001                        HLT+1
  (1)  011600  017701  167514                MOV     @CSR,R1         ;SAVE CSR
  (1)  011604  022701  100405                CMP     #100405,R1      ;CHECK CSR
  (1)  011610  001401                        BEQ     .+4             ;BRANCH IF OK
  (1)  011612  104001                        HLT+1                   ;CSR ERROR. POSSIBILITES:
  (1)                                                                ;(1) DONE DIDN'T CLEAR
  (1)
  (1)                                                                ;(2) TRANSMITTER UART DIDN'T CLR.
  (1)                                                                ;R1=CONTENTS OF CSR
  (1)
  (1)  011614  013700  001346                MOV     TIMER,  R0      ;SET UP TIMER
  (1)  011620  105305                4$:     DECB    R5              ;SHORT WAIT LOOP
  (1)  011622  001376                        BNE     4$
  (1)
  (1)  011624  017701  167470                MOV     @CSR,R1         ;SAVE CSR
  (1)  011630  105701                        TSTB    R1              ;CHECK FOR DONE
  (1)  011632  100001                        BPL     .+4             ;BRANCH IF OK
  (1)  011634  104001                        HLT+1                   ;DONE CAME UP!
  (1)                                                                ;MOS MUST NOT HAVE CLEARED
  (1)  011636  005300                        DEC     R0              ;TIMER COUNT
  (1)  011640  001367                        BNE     4$
```

```
(1)
(1)  011642  022701  100405                    CMP     #100405,R1      ;CHECK CSR
(1)  011646  001401                            BEQ     .+4             ;BRANCH IF OK
(1)  011650  104001                            HLT+1                   ;CSR ERROR
(1)
(1)  011652  017701  167444                    MOV     @RBUF,R1        ;CHECK RBUF
(1)  011656  100001                            BPL     .+4             ;BRANCH IF OK
(1)  011660  104001                            HLT+1                   ;RBUF NOT EMPTY!
(1)
(1)  011662  005077  167436                    CLR     @TCR
(1)  011666  005077  167426                    CLR     @CSR
(1)
(1)  011672  104400                            SCOPE
(1)
 4621
(1)                                    ;*************************************************************************
(1)                                    ;TEST 53:       TEST THAT TRANSMITTER READY CLEARS WHEN TBUF IS LOADED
(1)                                    ;       NOTE:   DUE TO THE DOUBLE BUFFERING BY THE UART, TWO CHARACTERS
(1)                                    ;                 MUST BE LOADED TO INSURE SEEING TRANSMITTER READY CLEAR
(1)                                    ;PROBABLE FAULTY LOGIC:  M7285 (D2-6) E39, E23, E49
(1)                                    ;*************************************************************************
(1)
(1)                                    ;               INITIALIZE
(1)                                    ;       DEVICE"CSR"REGISTER
(1)                                    ;                                       SET:
(1)  011674  004737  015420            TST53:  JSR     PC,@#INITD              ;BIT2 = MAINTENANCE
(1)                                                                            ;BIT3 = CLEAR MOS
(1)                                                                            ;BIT8 = MASTER XMTR SCAN ENB
(1)                                    ;WAIT FOR MOS TO CLEAR                  SET:
(1)                                    ;                                       ;BIT0 = RECEIVER ENABLE
(1)                                    ;
(1)                                    ;
(1)  011700  052777  000001  167416            BIS     #BIT0,  @TCR    ;TRANS CONTROL, LINE 0
(1)  011706  017701  167406           1$:      MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  011712  100375                            BPL     1$
(1)  011714  012777  000001  167404            MOV     #1,     @TBUF   ;TRANSMIT A 1
(1)  011722  017701  167372           2$:      MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  011726  100375                            BPL     2$
(1)  011730  012777  000002  167370            MOV     #2,     @TBUF   ;TRANSMIT A 2
(1)  011736  017701  167356                    MOV     @CSR,   R1      ;CHECK FOR XMTR READY
(1)  011742  100001                            BPL     .+4             ;BRANCH IF XMTR READY CLEARED
(1)  011744  104001                            HLT+1                   ;TRANSMITTER READY FAILED TO CLEAR
(1)                                                                    ;R1 = CONTENTS OF CSR
(1)  011746  013700  001346                    MOV     TIMER,  R0      ;SET UP TIMER
(1)  011752  105305                   3$:      DECB    R5              ;SHORT WAIT LOOP
(1)  011754  001376                            BNE     3$
(1)  011756  017701  167336                    MOV     @CSR,   R1      ;SAVE CSR FOR THE RECORD
(1)  011762  000400                            BR      .+2             ;NOP FOR TIMING
(1)  011764  005300                            DEC     R0              ;TIMER COUNT
(1)  011766  001371                            BNE     3$              ;BRANCH IF MORE TIME
(1)
(1)  011770  042701  000200                    BIC     #BIT7,R1        ;DONE MAY OR MAY NOT BE SET SO DONT TEST FOR IT
(1)  011774  022701  100405                    CMP     #100405,R1      ;TEST REST OF THE CSR
(1)  012000  001401                            BEQ     .+4
(1)  012002  104001                            HLT+1                   ;A CSR ERROR
(1)                                                                    ;R1 = CONTENTS OF CSR
```

```
(1)
(1)  012004  017701  167312          MOV     @RBUF,  R1      ;CHECK RBUF FOR CHAR PRESENT
(1)  012010  100401                  BMI     .+4             ;BRANCH IF CHAR PRESENT
(1)  012012  104001                  HLT+1                   ;CHAR PRESENT MISSING
(1)                                                          ;R1 = CONTENTS OF RBUF
(1)  012014  022701  100001          CMP     #100001,R1      ;CHECK THE DATA
(1)  012020  001401                  BEQ     .+4             ;BRANCH IF OK
(1)  012022  104001                  HLT+1                   ;RECEIVER ERROR
(1)                                                          ;R1 = CNTENTS OF RBUF
(1)  012024  017701  167272          MOV     @RBUF,  R1      ;CHECK RBUF FOR SECOND CHAR
(1)  012030  100401                  BMI     .+4             ;BRANCH IF CHAR PRESENT
(1)  012032  104001                  HLT+1                   ;CHAR PRESENT MISSING
(1)                                                          ;R1 = CONTENTS OF RBUF
(1)  012034  022701  100002          CMP     #100002,R1      ;CHECK THE DATA
(1)  012040  001401                  BEQ     .+4             ;BRANCH IF OK
(1)  012042  104001                  HLT+1                   ;RECEIVER ERROR
(1)                                                          ;R1 = CNTENTS OF RBUF
(1)  012044  017701  167252          MOV     @RBUF,  R1      ;CHECK FOR NO MORE CHARACTERS
(1)  012050  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT CLEARED
(1)  012052  104001                  HLT+1                   ;CHAR PRESENT NOT CLEAR!
(1)                                                          ;R1 = CONTENTS OF RBUF
(1)  012054  017701  167240          MOV     @CSR,   R1      ;SAVE CSR
(1)  012060  022701  100405          CMP     #100405,R1      ;CHECK CSR
(1)  012064  001401                  BEQ     .+4             ;BRANCH IF OK
(1)  012066  104001                  HLT+1                   ;CSR ERROR
(1)                                                          ;R1 = CONTENTS OF CSR
(1)  012070  005077  167230          CLR     @TCR            ;CLEAR TCR
(1)  012074  005077  167220          CLR     @CSR            ;CLEAR CSR
(1)  012100  104400                  SCOPE
(1)
4622
(1)                                  ;*************************************************************************
(1)                                  ;TEST 54:       TEST THE SILO ALARM LEVEL AT WHICH DONE WILL SET.
(1)                                  ;               TEST THAT RECEIVER ENABLE ON A 0 INHIBITS
(1)                                  ;               CHARACTER PRESENT.
(1)                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) E32, E15; M7279 (D11-3) E19,E23
(1)                                  ;*************************************************************************
(1)                                  ;               INITIALIZE
(1)                                  ;       DEVICE"CSR"REGISTER
(1)                                  ;                                               SET:
(1)  012102  004737  015420  000000  TST54:  JSR     PC,@#INITD      ;BIT2 = MAINTENANCE
(1)                                                                  ;BIT3 = CLEAR MOS
(1)                                                                  ;BIT8 = MASTER XMTR SCAN ENB
(1)                                  ;WAIT FOR MOS TO CLEAR
(1)                                  ;                               SET:
(1)                                                                  ;BIT0 = RECEIVER ENABLE
(1)  012106  052777  000001  167210          BIS     #BIT0,  @TCR    ;TRANS CONTROL, LINE 0
(1)  012114  005737  001350                  TST     ALMFLG          ;HAS THE ALARM LEVEL BEEN TESTED?
(1)  012120  001005                          BNE     11$             ;YES
(1)  012122  004737  015464                  JSR     PC,ALMCK        ;NO, GO DO IT
(1)  012126  012737  000001  001350          MOV     #1,ALMFLG       ;SET THE FLAG
(1)  012134  012777  000414  167156  11$:    MOV     #414,   @CSR    ;BIT2 = MAINTENANCE
(1)                                                                  ;BIT3 = CLEAR MOS
(1)                                                                  ;BIT8 = MASTER TRAN SCAN ENB
(1)  012142  032777  000020  167150  10$:    BIT     #BIT4,  @CSR    ;WAIT FOR MOS TO CLEAR
(1)  012150  001374                          BNE     10$
```

```
(1)                                     ;OUTPUT THE # OF CHARACTERS NECESSARY FOR THE SILO ALARM
(1)                                     ;LEVEL TO ALOW DONE TO SET.
(1)  012152  013702  001356             MOV     COUNT,R2        ;SET CHAR COUNT
(1)  012156  017701  167136     1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  012162  100375                     BPL     1$
(1)  012164  012777  000252  167134     MOV     #252,   @TBUF   ;SEND AN "*"
(1)  012172  005302                     DEC     R2
(1)  012174  001370                     BNE     1$
(1)  012176  013700  001346             MOV     TIMER,  R0      ;SET UP TIMER
(1)  012202  105305             2$:     DECB    R5              ;SHORT WAIT LOOP
(1)  012204  001376                     BNE     2$
(1)  012206  017701  167106             MOV     @CSR,   R1      ;SAVE CSR FOR TYPING
(1)  012212  105701                     TSTB    R1              ;CHECK FOR DONE
(1)  012214  100002                     BPL     3$              ;BRANCH IF NOT SET
(1)  012216  104001                     HLT+1                   ;DONE SET WHEN RCV ENB CLR
(1)                                                             ;R1=CONTENTS OF CSR
(1)  012220  000402                     BR      4$
(1)  012222  005300             3$:     DEC     R0              ;TIMER COUNT
(1)  012224  001366                     BNE     2$              ;BRANCH IF MORE TIMER
(1)
(1)  012226  017701  167070     4$:     MOV     @RBUF,  R1      ;CHECK AND SAVE FI/FO
(1)  012232  100001                     BPL     .+4             ;BRANCH IF OK
(1)  012234  104001                     HLT+1                   ;CHARACTER PRESENT IN FI/FO
(1)                                                             ;R1=DATA FROM FI/FO
(1)  012236  005277  167056             INC     @CSR            ;SET RECEIVER ENABLE
(1)  012242  017701  167054     5$:     MOV     @RBUF,  R1      ;CHECK FOR CHARACTER PRESENT
(1)  012246  100403                     BMI     6$              ;BRANCH IF OK
(1)  012250  105300                     DECB    R0              ;SHORT TIMER
(1)  012252  001373                     BNE     5$
(1)  012254  104001                     HLT+1                   ;CHARACTER PRESENT MISSING
(1)                                                             ;R1 = CONTENTS OF RBUF
(1)  012256  005077  167042     6$:     CLR     @TCR            ;CLR TRANS CONTROL REG
(1)
(1)  012262  104400                     SCOPE
4623
(1)
(1)                                     ;*****************************************************************************
(1)                                     ;TEST 55:       TEST THAT HALF DUPLEX (BIT1) DISABLES THE RECEIVER UARTS.
(1)                                     ;PROBABLE FAULTY LOGIC:  M7285 (D2-4) E32, E17, E22, (D2-2) E5, E1
(1)                                     ;*****************************************************************************
(1)
(1)  012264  004737  015410     TST55:  JSR     PC,     @#INITC ;INITIALIZE
(1)                                                             ;BIT1 = HALF DUPLEX
(1)                                                             ;BIT2 = MAINTENANCE
(1)                                                             ;BIT3 = CLEAR MOS
(1)                                                             ;BIT8 = MASTER TRAN SCAN ENB
(1)                                                             ;WAIT FOR MOS TO CLEAR
(1)                                                             ;BIT0 = RECEIVER ENABLE
(1)  012270  012777  000001  167026             MOV     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE0
(1)                                     ;OUTPUT THE # OF CHARACTERS NECESSARY FOR THE SILO ALARM
(1)                                     ;LEVEL TO ALOW DONE TO SET.
(1)  012276  013702  001356             MOV     COUNT,R2        ;SET CHAR COUNT
(1)  012302  017701  167012     1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  012306  100375                     BPL     1$
(1)  012310  012777  000252  167010     MOV     #252,   @TBUF   ;SEND AN "*"
(1)  012316  005302                     DEC     R2
(1)  012320  001370                     BNE     1$
```

```
(1)  012322  013700  001346                    MOV     TIMER,  R0      ;SET UP TIMER
(1)  012326  105305                    2$:     DECB    R5              ;SHORT WAIT LOOP
(1)  012330  001376                            BNE     2$
(1)  012332  017701  166762                    MOV     @CSR,   R1      ;SAVE CSR
(1)  012336  105701                            TSTB    R1              ;CHECK FOR DONE
(1)  012340  100002                            BPL     3$              ;BRANCH IF NOT SET
(1)  012342  104001                            HLT+1                   ;DONE SET WHEN HALF DUPLEX (BIT1) SET
(1)                                                                    ;R1=CONTENTS OF CSR
(1)  012344  000402                            BR      4$
(1)  012346  005300                    3$:     DEC     R0              ;TIMER COUNT
(1)  012350  001366                            BNE     2$              ;BRANCH IF MORE TIMER
(1)
(1)  012352  017701  166744            4$:     MOV     @RBUF,  R1      ;CHECK AND SAVE FI/FO
(1)  012356  100001                            BPL     .+4             ;BRANCH IF OK
(1)  012360  104001                            HLT+1                   ;CHARACTER PRESENT IN FI/FO
(1)                                                                    ;R1=DATA FROM FI/FO
(1)  012362  042777  000002  166730            BIC     #BIT1,  @CSR    ;CLEAR HALF DUPLEX BIT
(1)  012370  000240                            NOP
(1)  012372  000240                            NOP
(1)  012374  000240                            NOP
(1)  012376  000240                            NOP
(1)  012400  017701  166716            5$:     MOV     @RBUF,  R1      ;CHECK FOR CHAR PRESENT
(1)  012404  100001                            BPL     .+4             ;BRANCH IF CHAR NOT PRESENT
(1)  012406  104001                            HLT+1                   ;CHAR PRESENT AFTER H/D CLEARED
(1)                                                                    ;R1 = CONTENTS OF RBUF
(1)  012410  005077  166710                    CLR     @TCR            ;CLR TRANS CONTROL REG
(1)
(1)  012414  104400                            SCOPE
(1)
4624
(1)                                    ;*************************************************************************
(1)                                    ;TEST 56:       TEST THAT RECEIVER INTERRUPT DOES NOT OCCUR AT LEVEL 5
(1)                                    ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                                    ;*************************************************************************
(1)
(1)  012416  012777  012514  166704    TST56:  MOV     #ISR56, @RCVVEC ;SET UP XMTR INTERRUPT VECTOR
(1)  012424  012777  000340  166700            MOV     #340,   @RCVLVL ;AT LEVEL 7
(1)  012432  042737  000340  177776            BIC     #340,   @#PS    ;CLEAR PS LEVEL
(1)  012440  052737  000240  177776            BIS     #240,   @#PS    ;SET PS TO LEVEL 5
(1)  012446  004737  015400                    JSR     PC,     @#INITB ;SET:
(1)                                                                    ;BIT2 = MAINTENANCE
(1)                                                                    ;BIT3 = CLEAR MOS
(1)                                                                    ;BIT6 = RECEIVER INTERUPT ENABLE
(1)                                                                    ;BIT8 = MASTER TRANS SCAN ENABLE
(1)                                                                    ;WAIT FOR MOS TO CLEAR
(1)                                                                    ;BIT0 = RECEIVER ENABLE
(1)  012452  012777  000001  166644            MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)                                    ;OUTPUT THE # OF CHARACTERS NECESSARY FOR THE SILO ALARM
(1)                                    ;LEVEL TO ALOW DONE TO SET.
(1)  012460  013702  001356                    MOV     COUNT,R2        ;SET CHAR COUNT
(1)  012464  017701  166630            1$:     MOV     @CSR,   R1      ;WAIT FOR TRAN RDY
(1)  012470  100375                            BPL     1$
(1)  012472  012777  000025  166626            MOV     #25,    @TBUF   ;SEND #25
(1)  012500  005302                            DEC     R2
(1)  012502  001370                            BNE     1$
(1)  012504  105777  166610            2$:     TSTB    @CSR            ;WAIT FOR DONE
```

```
(1)  012510  100375                        BPL     2$
(1)  012512  000404                        BR      END56           ;OK, BRANCH IF NO INTERRUPT
(1)
(1)  012514  104000              ISR56:     HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 5
(1)  012516  012716  012524                MOV     #END56,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  012522  000002                        RTI
(1)
(1)  012524  017701  166572      END56:    MOV     @RBUF,  R1      ;READ THE CHARACTER
(1)  012530  100401                        BMI     .+4             ;BRANCH IF CHAR PRESENT
(1)  012532  104001                        HLT+1                   ;CHAR PRESENT MISSING
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)  012534  022701  100025                CMP     #100025,R1      ;CHECK THE DATA
(1)  012540  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  012542  104001                        HLT+1                   ;RECEIVED DATA ERROR
(1)                                                                ;R1 = CONTENTS OF RBUF
(1)  012544  013777  001332  166556        MOV     RCVLVL,@RCVVEC
(1)  012552  012777  000004  166552        MOV     #IOT,   @RCVLVL
(1)  012560  005077  166540                CLR     @TCR
(1)  012564  005077  166530                CLR     @CSR
(1)  012570  042737  000340  177776        BIC     #340,@#PS
(1)
(1)  012576  104400                        SCOPE
(1)
4625
(1)                              ;*************************************************************************
(1)                              ;TEST 57:       TEST THAT RECEIVER INTERRUPT OCCURES AT LEVEL 4
(1)                              ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
(1)                              ;*************************************************************************
(1)
(1)  012600  012777  012700  166522  TST57:   MOV     #ISR57, @RCVVEC ;SET UP XMTR INTERRUPT VECTOR
(1)  012606  012777  000340  166516           MOV     #340,   @RCVLVL ;AT LEVEL 7
(1)  012614  042737  000340  177776           BIC     #340,   @#PS    ;CLEAR PS LEVEL
(1)  012622  052737  000200  177776           BIS     #200,   @#PS    ;SET PS TO LEVEL 4
(1)  012630  004737  015400                   JSR     PC,     @#INITB ;SET:
(1)                                                                   ;BIT2 = MAINTENANCE
(1)                                                                   ;BIT3 = CLEAR MOS
(1)                                                                   ;BIT6 = RECEIVER INTERUPT ENABLE
(1)                                                                   ;BIT8 = MASTER TRANS SCAN ENABLE
(1)                                                                   ;WAIT FOR MOS TO CLEAR
(1)                                                                   ;BIT0 = RECEIVER ENABLE
(1)  012634  012777  000001  166462           MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
(1)                              ;OUTPUT THE # OF CHARACTERS NECESSARY FOR THE SILO ALARM
(1)                              ;LEVEL TO ALOW DONE TO SET.
(1)  012642  013702  001356                   MOV     COUNT,R2        ;SET CHAR COUNT
(1)  012646  017701  166446           1$:     MOV     @CSR,   R1      ;WAIT FOR TRAN RDY
(1)  012652  100375                           BPL     1$
(1)  012654  012777  000025  166444           MOV     #25,    @TBUF   ;SEND #25
(1)  012662  005302                           DEC     R2
(1)  012664  001370                           BNE     1$
(1)  012666  105777  166426           2$:     TSTB    @CSR            ;WAIT FOR DONE
(1)  012672  100375                           BPL     2$
(1)  012674  104000                           HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 4
(1)  012676  000403                           BR      END57           ;CONTINUE
(1)
(1)  012700  012716  012706           ISR57:  MOV     #END57,(SP)     ;MOVE NEW RTI ADR ONTO STACK
(1)  012704  000002                           RTI
```

```
(1)
(1)  012706  017701  166410          END57:  MOV     @RBUF,  R1      ;READ THE CHARACTER
(1)  012712  100401                         BMI     .+4             ;BRANCH IF CHAR PRESENT
(1)  012714  104001                         HLT+1                   ;CHAR PRESENT MISSING
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)  012716  022701  100025                 CMP     #100025,R1      ;CHECK THE DATA
(1)  012722  001401                         BEQ     .+4             ;BRANCH IF OK
(1)  012724  104001                         HLT+1                   ;RECEIVED DATA ERROR
(1)                                                                 ;R1 = CONTENTS OF RBUF
(1)  012726  013777  001332  166374         MOV     RCVLVL,@RCVVEC
(1)  012734  012777  000004  166370         MOV     #IOT,   @RCVLVL
(1)  012742  005077  166356                 CLR     @TCR
(1)  012746  005077  166346                 CLR     @CSR
(1)  012752  042737  000340  177776         BIC     #340,@#PS
(1)
(1)  012760  104400                         SCOPE
(1)
 4626
(1)                                 ;******************************************************************************
(1)                                 ;TEST 60:       TEST FI/FO OVERRUN
(1)                                 ;               THE FI/FO BUFFER SHOULD HOLD 64 CHARACTERS.
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D1-7) E32, E17, E22 (D2-2) E5, E1
(1)                                 ;******************************************************************************
(1)
(1)                                 ;               INITIALIZE
(1)                                 ;       DEVICE"CSR"REGISTER
(1)                                 ;                                               SET:
(1)  012762  004737  015420         TST60:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
(1)                                                                 ;BIT3 = CLEAR MOS
(1)                                                                 ;BIT8 = MASTER XMTR SCAN ENB
(1)                                 ;WAIT FOR MOS TO CLEAR
(1)                                 ;                               SET:
(1)                                 ;                               ;BIT0 = RECEIVER ENABLE
(1)                                 ;
(1)  012766  012777  177777  166330         MOV     #177777,@TCR    ;TRANS CONTROL BIT, ALL LINES
(1)  012774  012700  000100                 MOV     #100,   R0      ;SET UP COUNTER - 64. CHAR FI/FO BUFF
(1)  013000  017701  166314         1$:     MOV     @CSR,   R1      ;SAVE AND WAIT FOR TRANS READY
(1)  013004  100375                         BPL     1$
(1)  013006  000377  166314                 SWAB    @TBUF           ;TRANSMIT LINE # ON LINE
(1)  013012  032701  020000                 BIT     #BIT13, R1      ;CHECK FI/FO OVERRUN
(1)  013016  001401                         BEQ     .+4             ;BRANCH IF OK
(1)  013020  104001                         HLT+1                   ;FI/FO OVERRUN TOO SOON
(1)                                                                 ;R1=CONTENTS OF CSR
(1)  013022  005300                         DEC     R0              ;COUNT DOWN
(1)  013024  001365                         BNE     1$
(1)
(1)  013026  013700  001346                 MOV     TIMER,  R0      ;SET UP TIMER
(1)  013032  017701  166262         2$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
(1)  013036  100375                         BPL     2$
(1)  013040  105305         3$:     DECB    R5                      ;SHORT WAIT LOOP
(1)  013042  001376                         BNE     3$
(1)  013044  017701  166250                 MOV     @CSR,   R1      ;SAVE CSR FOR THE RECORD
(1)  013050  100401                         BMI     .+4             ;BRANCH IF TRANS READY
(1)  013052  104001                         HLT+1                   ;TRANS READY MISTERIOUSLY DISAPPEARED
(1)                                                                 ;R1=CONTENTS OF CSR
(1)  013054  005300                         DEC     R0              ;TIME 3 CHARACTER LENGTHS
```

```
(1)  013056  001370                    BNE     3$              ;BRANCH IF MORE TIME
(1)
(1)                                  ;FI/FO SHOULD NOW BE FULL
(1)
(1)  013060  105701                    TSTB    R1              ;CHECK THAT DONE IS SET
(1)  013062  100401                    BMI     .+4             ;BRANCH IF OK
(1)  013064  104001                    HLT+1                   ;DONE DIDN'T COME UP!
(1)                                                            ;R1 = CONTENTS OF CSR
(1)  013066  022701  100605            CMP     #100605,R1      ;CHECK THAT FI/FO NOT OVERRUN
(1)  013072  001401                    BEQ     .+4             ;BRANCH IF OK
(1)  013074  104001                    HLT+1                   ;FI/FO OVERRUN SET
(1)                                                            ;OR SOME OTHER CSR ERROR
(1)                                                            ;R1=CONTENTS OF CSR
(1)
(1)                                  ;******************************************************************************
(1)                                  ;TEST 60A:      TEST THAT FI/FO OVERRUN COMES UP WHEN 65TH CHARACTER
(1)                                  ;                 IS RECEIVED WITHOUT READING FI/FO
(1)                                  ;******************************************************************************
(1)
(1)  013076  000377  166224          T60A:   SWAB    @TBUF           ;SEND 65TH CHARACTER
(1)  013102  013700  001346                  MOV     TIMER,  R0      ;SET UP TIMER
(1)  013106  105305                  11$:    DECB    R5              ;SHORT WAIT LOOP
(1)  013110  001376                          BNE     11$
(1)  013112  017701  166202                  MOV     @CSR,R1         ;SAVE CSR
(1)  013116  032701  020000                  BIT     #BIT13,R1       ;CHECK FI/FO OVERRUN
(1)  013122  001003                          BNE     12$             ;BRANCH WHEN SET
(1)  013124  005300                          DEC     R0              ;TIMER
(1)  013126  001367                          BNE     11$             ;BRANCH IF MORE TIME
(1)  013130  104001                          HLT+1                   ;FI/FO OVERRUN DIDN'T COMEUP
(1)                                                                  ;R1 = CONTENTS OF CSR
(1)  013132  022701  120605          12$:    CMP     #120605,R1      ;CHECK TOTAL CSR
(1)  013136  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  013140  104001                          HLT+1                   ;SOMETHING IN CSR FOULED UP
(1)                                                                  ;R1=CONTENTS OF CSR
(1)
(1)                                  ;******************************************************************************
(1)                                  ;TEST 60B:      TEST THAT READING THE RECEIVER BUFFER CAUSES FI/FO
(1)                                  ;                 OVERRUN TO CLEAR.
(1)                                  ;       NOTE:   BECAUSE OF TIMING OF THE FI/FO, FI/FO OVERRUN CAN COME
(1)                                  ;                 BACK UP AFTER READING ONE CHARACTER, SO A SECOND MUST
(1)                                  ;                 BE READ TO INSURE THAT FI/FO OVERRUN IS CLEAR.
(1)                                  ;******************************************************************************
(1)
(1)  013142  012700  000002          T60B:   MOV     #2,     R0      ;SET UP COUNTER - 2 CHARACTERS
(1)  013146  017701  166150          21$:    MOV     @RBUF,  R1      ;CHECK AND SAVE FIRST CHAR IN FI/FO
(1)  013152  100401                          BMI     .+4             ;BRANCH IF CHAR PRESENT
(1)  013154  104001                          HLT+1                   ;CHARACTER PRESENT GONE!
(1)
(1)  013156  032701  070000                  BIT     #070000,R1      ;CHECK RECEIVER ERRORS
(1)  013162  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  013164  104001                          HLT+1                   ;RECEIVER  ERROR
(1)                                                                  ;R1=CONTENTS OF RBUF
(1)                                                                  ;BIT14=UART OVERRUN
(1)                                                                  ;BIT13=FRAMMING ERROR
(1)                                                                  ;BIT12=PARITY ERROR
(1)
```

```
MAINDEC-11-DZDJA-E     DJ11 LOGIC TESTS          MACY11 30(1046)  08-SEP-77  15:09  PAGE 53-43
DZDJAE.P11    08-SEP-77 15:06          TST60:  TEST FI/FO OVERRUN                              SEQ 0063

   (1)  013166  010102                        MOV     R1,     R2      ;PUT LINE # IN R2
   (1)  013170  000302                        SWAB    R2
   (1)  013172  042702  177760                BIC     #177760,R2
   (1)  013176  120102                        CMPB    R1,     R2      ;CHECK DATA (=LINE#)
   (1)  013200  001401                        BEQ     .+4             ;BRANCH IF OK
   (1)  013202  104001                        HLT+1                   ;WRONG DATA RECEIVED
   (1)                                                                ;R1=FI/FO DATA
   (1)                                                                ;(DATA SHOULD=LINE#)
   (1)  013204  005300                22$:     DEC     R0              ;COUNT CHARACTERS
   (1)  013206  001403                        BEQ     24$             ;BRANCH WHEN DONE
   (1)  013210  105305                23$:     DECB    R5              ;SHORT WAIT LOOP - GIVE FI/FO TIME
   (1)  013212  001376                        BNE     23$
   (1)  013214  000754                        BR      21$             ;GO READ ANOTHER
   (1)
   (1)  013216  017701  166076        24$:     MOV     @CSR,   R1      ;SAVE CSR
   (1)  013222  022701  100605                CMP     #100605,R1      ;CHECK THAT FI/FO OVERRUN CLEARED
   (1)  013226  001401                        BEQ     .+4             ;BRANCH IF OK
   (1)  013230  104001                        HLT+1                   ;FI/FO OVERRUN DIDN'T CLR
   (1)                                                                ;OR SOMEOTHER CSR PROBLEM
   (1)                                                                ;R1=CONTENTS OF CSR
   (1)
   (1)                                ;*****************************************************************
   (1)                                ;TEST 60C:      TEST THAT FI/FO OVERRUN INTERRUPT
   (1)                                ;               DOESN'T OCCUR WHEN THE PROCESSOR IS AT LEVEL 5
   (1)                                ;*****************************************************************
   (1)
   (1)  013232  042737  000340  177776  T60C:  BIC     #340,   @#PS    ;CLEAR PSW
   (1)  013240  052737  000240  177776        BIS     #240,   @#PS    ;SET PROCESSOR TO LEVEL 5
   (1)  013246  012777  013336  166054        MOV     #32$,   @RCVVEC ;SET UP RECEIVER INTERRUPT VEC
   (1)  013254  012777  000340  166050        MOV     #340,   @RCVLVL
   (1)  013262  052777  010000  166030        BIS     #BIT12, @CSR    ;SET STATUS ENABLE
   (1)  013270  017701  166024                MOV     @CSR,   R1      ;SAVE CSR
   (1)  013274  022701  110605                CMP     #110605,R1      ;CHECK CSR
   (1)  013300  001401                        BEQ     .+4
   (1)  013302  104001                        HLT+1                   ;CSR ERROR
   (1)                                                                ;R1=CONTENTS OF CSR
   (1)  013304  000377  166016                SWAB    @TBUF           ;SEND LINE #
   (1)  013310  005777  166004        30$:    TST     @CSR            ;WAIT FOR TRANSMITTER READY
   (1)  013314  100375                        BPL     30$
   (1)  013316  000377  166004                SWAB    @TBUF           ;SEND LINE #
   (1)  013322  017701  165772        31$:    MOV     @CSR,   R1      ;SAVE CSR
   (1)  013326  032701  020000                BIT     #BIT13, R1      ;WAIT FOR FI/FO OVERRUN
   (1)  013332  001773                        BEQ     31$
   (1)  013334  000410                        BR      33$             ;SKIP ISR
   (1)
   (1)  013336  017701  165756        32$:    MOV     @CSR,   R1      ;SAVE CSR
   (1)  013342  104001                        HLT+1                   ;INTERRUPT OCCURRED AT LEVEL 5
   (1)  013344  005777  165752                TST     @RBUF           ;"POP" ONE CHARACTER
   (1)  013350  012716  013362                MOV     #34$,   (SP)    ;RESET RETURN ADDRESS
   (1)  013354  000002                        RTI                     ;RETURN
   (1)
   (1)  013356  012700  000002        33$:    MOV     #2,     R0      ;SET UP COUNTER - 2 CHARACTERS
   (1)  013362  017701  165734        34$:    MOV     @RBUF,  R1      ;READ ONE CHARACTER
   (1)  013366  100401                        BMI     .+4             ;BRANCH IF CHARACTER PRESENT
   (1)  013370  104001                        HLT+1
```

```
(1)
(1)  013372  032701  070000            BIT     #70000, R1      ;CHECK ERRORS
(1)  013376  001401                     BEQ     .+4
(1)  013400  104001                     HLT+1
(1)
(1)  013402  010102                     MOV     R1,     R2      ;DUP DATA
(1)  013404  000302                     SWAB    R2
(1)  013406  042702  177760             BIC     #177760,R2      ;CLR ALL BUT LINE #
(1)  013412  120102                     CMPB    R1,     R2      ;CHECK DATA
(1)  013414  001401                     BEQ     .+4
(1)  013416  104001                     HLT+1
(1)
(1)  013420  005300                     DEC     R0              ;COUNT CHARACTERS
(1)  013422  003403                     BLE     36$             ;BRANCH IF DONE
(1)  013424  105305            35$:     DECB    R5              ;SHORT WAIT LOOP
(1)  013426  001376                     BNE     35$
(1)  013430  000754                     BR      34$             ;GO READ ANOTHER CHARACTER
(1)
(1)  013432  017701  165662      36$:   MOV     @CSR,   R1      ;SAVE CSR
(1)  013436  022701  110605             CMP     #110605,R1      ;CHECK THAT FI/FO OVERRUN CLEARED
(1)  013442  001401                     BEQ     .+4             ;BRANCH IF OK
(1)  013444  104001                     HLT+1                   ;FI/FO OVERRUN DIDN'T CLR
(1)                                                             ;OR SOMEOTHER CSR PROBLEM
(1)                                                             ;R1=CONTENTS OF CSR
(1)
(1)                             ;******************************************************************************
(1)                             ;TEST 60D:      TEST THAT FI/FO OVERRUN INTERRUPT
(1)                             ;               OCCURS WHEN PROCESSOR IS AT LEVEL 4
(1)                             ;******************************************************************************
(1)
(1)  013446  042737  000340  177776  T60D:  BIC     #340,   @#PS    ;CLEAR PROCESSOR LEVEL
(1)  013454  052737  000200  177776        BIS     #200,   @#PS    ;SET PROCESSOR TO LEVEL 4
(1)  013462  012777  013546  165640        MOV     #42$,   @RCVVEC ;SET UP RECEIVER INTERRUPT VEC
(1)  013470  017701  165624               MOV     @CSR,   R1      ;SAVE CSR
(1)  013474  022701  110605               CMP     #110605,R1      ;CHECK CSR
(1)  013500  001401                       BEQ     .+4
(1)  013502  104001                       HLT+1
(1)
(1)  013504  000377  165616               SWAB    @TBUF           ;SEND LINE #
(1)  013510  005777  165604        40$:   TST     @CSR            ;WAIT FOR TRANSMITTER READY
(1)  013514  100375                       BPL     40$
(1)  013516  000377  165604               SWAB    @TBUF           ;SEND LINE #
(1)  013522  017701  165572        41$:   MOV     @CSR,   R1      ;SAVE CSR
(1)  013526  032701  020000               BIT     #BIT13, R1      ;WAIT FOR FI/FO OVERRUN
(1)  013532  001773                       BEQ     41$
(1)  013534  104001                       HLT+1                   ;INTERUPT DIDN'T OCCURE WHEN OVERRUN SET
(1)                                                               ;R1 = CONTENTS OF CSR
(1)  013536  052777  000010  165554       BIS     #BIT3,  @CSR    ;CLEAR MOS
(1)  013544  000470                       BR      45$             ;SKIP TO THE END
(1)
(1)  013546  017701  165546        42$:   MOV     @CSR,   R1      ;SAVE CSR
(1)  013552  022701  130605               CMP     #130605,R1      ;CHECK CSR
(1)  013556  001401                       BEQ     .+4             ;BRANCH IF OK
(1)  013560  104001                       HLT+1                   ;CSR ERROR
(1)                                                               ;R1 = CONTENTS OF CSR
(1)  013562  012700  000101               MOV     #101,   R0      ;SET UP COUNTER - 65 CHARACTERS
```

```
(1)  013566  017701  165530        43$:  MOV    @RBUF,  R1      ;READ ONE CHARACTER
(1)  013572  100401                      BMI    .+4             ;BRANCH IF CHARACTER PRESENT
(1)  013574  104001                      HLT+1
(1)
(1)  013576  032701  070000              BIT    #70000, R1      ;CHECK ERRORS
(1)  013602  001401                      BEQ    .+4
(1)  013604  104001                      HLT+1
(1)
(1)  013606  010102                      MOV    R1,     R2      ;DUP DATA
(1)  013610  000302                      SWAB   R2
(1)  013612  042702  177760              BIC    #177760,R2      ;CLR ALL BUT LINE #
(1)  013616  120102                      CMPB   R1,     R2      ;CHECK DATA
(1)  013620  001401                      BEQ    .+4
(1)  013622  104001                      HLT+1
(1)
(1)  013624  017701  165470              MOV    @CSR,   R1      ;SAVE CSR
(1)  013630  005300                      DEC    R0
(1)  013632  001420                      BEQ    46$             ;GET OUT IF ALL CHAR'S READ
(1)  013634  022700  000100              CMP    #100,   R0      ;CHECK FOR FIRST CHAR READ
(1)  013640  001752                      BEQ    43$             ;SKIP CSR CHECK ON FIRST CHAR
(1)  013642  020037  001356              CMP    R0,COUNT        ;IF CHAR'S LEFT IN SILO IS LESS THEN
(1)                                                             ;ALARM LEVEL, DONE WILL GO AWAY.
(1)  013646  002405                      BLT    44$
(1)  013650  022701  110605              CMP    #110605,R1      ;CHECK CSR
(1)  013654  001401                      BEQ    .+4
(1)  013656  104001                      HLT+1
(1)
(1)  013660  000742                      BR     43$
(1)
(1)  013662  022701  110405        44$:  CMP    #110405,R1      ;CHECK CSR
(1)  013666  001401                      BEQ    .+4             ;BRANCH IF OK
(1)  013670  104001                      HLT+1                  ;DONE DIDN'T GO AWAY
(1)                                                             ;OR OTHER CSR ERROR
(1)  013672  000735                      BR     43$
(1)
(1)  013674  017701  165422        46$:  MOV    @RBUF,  R1      ;READ A CHARACTER
(1)  013700  100001                      BPL    .+4             ;BRANCH IF NO CHAR. PRES
(1)  013702  104001                      HLT+1                  ;CHAR. PRESENT!
(1)
(1)  013704  013777  001332  165416      MOV    RCVLVL, @RCVVEC
(1)  013712  012777  000004  165412      MOV    #IOT,   @RCVLVL
(1)  013720  012716  013726              MOV    #45$,   (SP)    ;RESET RETURN ADDRESS ON STACK
(1)  013724  000002                      RTI                    ;RESTORE PSW
(1)  013726  005077  165372        45$:  CLR    @TCR
(1)  013732  005077  165362              CLR    @CSR
(1)  013736  042737  000340  177776      BIC    #340,   @#PS    ;LOWER PROCESSOR STATUS
(1)
(1)  013744  104400                      SCOPE
(1)
4627
(1)                              ;******************************************************************************
(1)                              ;TEST 61:       TEST THAT UART OVERRUN IS DETECTED ON ALL LINES
(1)                              ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E3; E1; M7279; M7280
(1)                              ;******************************************************************************
(1)
(1)  013746  012777  000414  165344  TST61:  MOV    #414,   @CSR    ;BIT2 = MAINTENANCE
```

```
(1)                                                                     ;BIT3 = CLEAR MOS
(1)                                                                     ;BIT8 = TRANS SCAN ENABLE
(1)  013754  032777  000020  165336  10$:    BIT     #BIT4,  @CSR     ;WAIT FOR MOS TO CLEAR
(1)  013762  001374                         BNE     10$
(1)  013764  005001                         CLR     R1               ;SET UP LINE COUNTER
(1)  013766  012777  000001  165330         MOV     #1,     @TCR     ;TRANS CONTROL, LINE 0
(1)  013774  017702  165320          LOP61: MOV     @CSR,   R2       ;WAIT FOR TRANS READY
(1)  014000  100375                         BPL     LOP61
(1)  014002  012777  000001  165316         MOV     #1,     @TBUF    ;SEND #1
(1)  014010  017702  165304          2$:    MOV     @CSR,   R2       ;WAIT FOR TRANS READY
(1)  014014  100375                         BPL     2$
(1)  014016  012777  000002  165302         MOV     #2,     @TBUF    ;SEND #2
(1)  014024  013700  001346                 MOV     TIMER,  R0       ;SET  UP TIMER
(1)  014030  105305                  3$:    DECB    R5               ;SHORT TIME LOOP
(1)  014032  001376                         BNE     3$
(1)  014034  017702  165260                 MOV     @CSR,   R2       ;SAVE CSR FOR THE RECORD
(1)  014040  000400                         BR      .+2              ;NOP FOR TIMING
(1)  014042  005300                         DEC     R0               ;TIMER COUNT
(1)  014044  001371                         BNE     3$
(1)  014046  005277  165246                 INC     @CSR             ;SET RECEIVER ENABLE
(1)  014052  017702  165242                 MOV     @CSR,   R2       ;SAVE CSR
(1)  014056  032702  000001                 BIT     #BIT0,  R2       ;CHECK RECEIVER ENABLE
(1)  014062  001001                         BNE     .+4              ;BRANCH IF OK
(1)  014064  104002                         HLT+2                    ;RECEIVER ENABLE FAILED TO SET
(1)                                                                     ;R1 = LINE #
(1)                                                                     ;R2 = CONTENTS OF CSR
(1)  014066  013700  001346                 MOV     TIMER,R0         ;SET UP TIMER
(1)  014072  105305                  4$:    DECB    R5               ;WAIT SHORT TIME LOOP
(1)  014074  001376                         BNE     4$
(1)  014076  017702  165220                 MOV     @RBUF,R2         ;CHECK FOR CHAR PRES
(1)  014102  100403                         BMI     5$               ;OK
(1)  014104  005300                         DEC     R0               ;TIMER COUNT
(1)  014106  001371                         BNE     4$
(1)  014110  104002                         HLT+2                    ;NO CHARACTER PRESENT
(1)                                                                     ;R1 = LINE #
(1)                                                                     ;R2 = CONTENTS OF RBUF
(1)  014112  032702  040000          5$:    BIT     #BIT14, R2       ;CHECK FOR UART OVERRRUN
(1)  014116  001001                         BNE     .+4              ;BRANCH IF OK
(1)  014120  104002                         HLT+2                    ;UART OVERRUN MISSING
(1)                                                                     ;R1 = LINE #
(1)                                                                     ;R2 = CONTENTS OF RBUF
(1)  014122  122702  000002                 CMPB    #2,     R2       ;CHECK THE DATA
(1)  014126  001401                         BEQ     .+4              ;BRANCH IF OK
(1)  014130  104002                         HLT+2                    ;DATA ERROR - 3RD CHAR OVERRUNS 2ND
(1)                                                                     ;R1 = LINE #
(1)                                                                     ;R2 = CONTENTS OF RBUF
(1)  014132  010203                         MOV     R2,     R3       ;DUP DATA
(1)  014134  000303                         SWAB    R3
(1)  014136  042703  177700                 BIC     #177700,R3       ;MASK ALL BIT LINE #, ERROR BITS
(1)  014142  020103                         CMP     R1,     R3       ;CHECK LINE #, ERRORS
(1)  014144  001401                         BEQ     .+4              ;BRANCH IF OK
(1)  014146  104002                         HLT+2                    ;LINE # OR OTHER RBUF ERROR
(1)                                                                     ;R1 = LINE #
(1)                                                                     ;R2 = CONTENTS OF RBUF
(1)  014150  017702  165146          7$:    MOV     @RBUF,  R2       ;CHECK FOR MORE DATA
(1)  014154  100002                         BPL     8$               ;BRANCH IF OK
```

```
(1)  014156  104002                          HLT+2                ;EXTRA DATA IN FI/FO!
(1)                                                                ;R1 = LINE #
(1)                                                                ;R2 = CONTENTS OF RBUF
(1)  014160  000773                          BR       7$
(1)
(1)  014162  005377  165132          8$:     DEC      @CSR        ;CLEAR RECEIVER ENABLE
(1)  014166  017702  165126                  MOV      @CSR,    R2 ;SAVE CSR
(1)  014172  022702  100404                  CMP      #100404,R2  ;CHECK CSR
(1)  014176  001401                          BEQ      .+4         ;BRANCH IF OK
(1)  014200  104002                          HLT+2                ;CSR ERROR
(1)                                                                ;R1 = LINE NUMBER
(1)                                                                ;R2 = CONTENTS OF CSR
(1)  014202  005201                          INC      R1          ;COUNT LINES
(1)  014204  006377  165114                  ASL      @TCR        ;GO TO NEXT LINE
(1)  014210  103271                          BCC      LOP61       ;BRANCH BACK IF MORE LINES
(1)
(1)  014212  005077  165102                  CLR      @CSR
(1)
(1)  014216  104400                          SCOPE
(1)
4628
(1)                                ;******************************************************************************
(1)                                ;TEST 62:       TEST BITS OF BCSR FOR READ/WRITE CAPABILITY
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5, E1, (D2-3) E16, E2, E19, E35
(1)                                ;******************************************************************************
(1)
(1)  014220  012777  002010  165072  TST62:  MOV      #002010,@CSR    ;SET CSR
(1)  014226  012777  177777  165070          MOV      #177777,@BCSR   ;SET ALL BITS OF BCSR
(1)  014234  017701  165064                  MOV      @BCSR,   R1     ;CHECK AND SAVE BCSR
(1)  014240  022701  177777                  CMP      #177777,R1      ;CHECK THAT ALL THE BITS ARE SET
(1)  014244  001401                          BEQ      .+4             ;BRANCH IF OK
(1)  014246  104001                          HLT+1                    ;BIT(S) OF BCSR FAILED TO SET
(1)
(1)  014250  005077  165050                  CLR      @BCSR           ;CLEAR BCSR
(1)  014254  017701  165044                  MOV      @BCSR,   R1     ;CHECK THAT IT CLEARED AND SAVE
(1)  014260  001401                          BEQ      .+4             ;BRANCH IF CLR
(1)  014262  104001                          HLT+1                    ;BIT(S) OF BCSR FAILED TO CLEAR
(1)
(1)  014264  104400                          SCOPE
4629
(1)                                ;******************************************************************************
(1)                                ;TEST 63:       TEST THAT LINE0 CAN TRANSMIT AND RECEIVE A BREAK
(1)                                ;         ALSO CHECKS FRAMING ERROR(RBUF BIT13)
(1)                                ;        ALSO CHECKS PARITY ERROR(RBUF BIT12)
(1)                                ;        IF ODD PARITY IS SELECTED
(1)                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5, E1, (D2-3) E16, E2, E19, E35
(1)                                ;******************************************************************************
(1)
(1)  014266  004737  015370  TST63:  JSR      PC,      @#INITA ;INITIALIZE
(1)                                                             ;BIT2 = MAINTENANCE
(1)                                                             ;BIT3 = CLEAR MOS
(1)                                                             ;BIT10= R/W BCSR
(1)                                                             ;WAIT FOR MOS TO CLEAR
(1)                                                             ;BIT0 = RECEIVER ENABLE
(1)  014272  012777  000001  165024          MOV      #1,      @BCSR   ;SEND BREAKS, LINE 0
(1)  014300  013700  001346                  MOV      TIMER,   R0      ;SET UP TIMER
```

```
(1)  014304  105305                  1$:   DECB   R5              ;SHORT WAIT LOOP
(1)  014306  001376                        BNE    1$
(1)  014310  017701  165006              MOV    @RBUF,  R1       ;SAVE CHAR PRES
(1)  014314  100403                        BMI    2$              ;BRANCH WHEN FOUND
(1)  014316  005200                        INC    R0              ;WAIT A WHILE
(1)  014320  001371                        BNE    1$
(1)  014322  104001                        HLT+1                  ;CHAR PRES NEVER CAME UP
(1)                                                               ;R1=CONTENTS OF RBUF
(1)
(1)  014324  032701  020000          2$:   BIT    #020000,R1      ;CHECK FOR FRAMING ERROR
(1)  014330  001001                        BNE    .+4             ;BRANCH IF OK
(1)  014332  104001                        HLT+1                  ;FRAMING ERROR NOT UP
(1)                                                               ;R1=CONTENTS OF RBUF
(1)  014334  133737  001366  001300        BITB   DJPAR,  PARITY  ;CHECK ODD PARITY FLAG
(1)  014342  001404                        BEQ    3$              ;BRANCH IF NOT
(1)  014344  032701  010000              BIT    #010000,R1      ;CHECK PARITY ERROR
(1)  014350  001005                        BNE    4$              ;
(1)  014352  104001                        HLT+1                  ;ODD PARITY SHOULD CAUSE PARITY ERROR
(1)                                                               ;R1=CONTENTS OF RBUF
(1)
(1)  014354  032701  010000          3$:   BIT    #010000,R1      ;CHECK PARITY ERROR
(1)  014360  001401                        BEQ    .+4             ;BRANCH IF OK
(1)  014362  104001                        HLT+1                  ;EVEN PARITY OR NO PARITY
(1)                                                               ;SHOULDN'T CAUSE PARITY ERROR
(1)                                                               ;R1=CONTENTS OF RBUF
(1)  014364  032701  040000          4$:   BIT    #040000,R1      ;CHECK UART OVERRUN
(1)  014370  001401                        BEQ    .+4             ;BRANCH IF OK
(1)  014372  104001                        HLT+1                  ;UART OVERRUN SET!!
(1)                                                               ;R1=CONTENTS OF RBUF
(1)  014374  032701  007400              BIT    #007400,R1      ;CHECK LINE #
(1)  014400  001401                        BEQ    .+4             ;BRANCH IF OK
(1)  014402  104001                        HLT+1                  ;WRONG LINE# IN FI/FO
(1)                                                               ;R1=CONTENTS OF RBUF
(1)  014404  105701                        TSTB   R1              ;CHECK DATA
(1)  014406  001401                        BEQ    .+4             ;BRANCH IF OK
(1)  014410  104001                        HLT+1                  ;WRONG DATA RECEIVED
(1)                                                               ;R1=CONTENTS OF RBUF
(1)
(1)  014412  017701  164704              MOV    @RBUF,  R1       ;READ FI/FO AGAIN
(1)  014416  100001                        BPL    .+4             ;BRANCH IF OK
(1)  014420  104001                        HLT+1                  ;EXTRA CHAR IN FI/FO
(1)                                                               ;R1 = CONTENTS OF RBUF
(1)  014422  005077  164676              CLR    @BCSR           ;CLEAR BREAK CONTROL REG
(1)  014426  105305                  5$:   DECB   R5              ;SHORT WAIT LOOP-REGISTER A MARK
(1)  014430  001376                        BNE    5$
(1)
(1)  014432  104400                        SCOPE
(1)
4630
(1)                                 ;*************************************************************************
(1)                                 ;TEST 64:       TEST THAT EACH LINE CAN TRANSMIT AND RECEIVE A BREAK
(1)                                 ;               ALSO CHECKS FRAMING ERROR(RBUF BIT13)
(1)                                 ;               ALSO CHECKS PARITY ERROR(RBUF BIT12)
(1)                                 ;                 IF ODD PARITY IS SELECTED
(1)                                 ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5, E1, (D2-3) E16, E2, E19, E35
(1)                                 ;*************************************************************************
```

```
(1)
(1)  014434  004737  015370          TST64:  JSR     PC,     @#INITA ;INITIALIZE
(1)                                                                 ;BIT2 = MAINTENANCE
(1)                                                                 ;BIT3 = CLEAR MOS
(1)                                                                 ;BIT10= R/W BCSR
(1)                                                                 ;WAIT FOR MOS TO CLEAR
(1)                                                                 ;BIT0 = RECEIVER ENABLE
(1)  014440  005001                          CLR     R1              ;SET UP LINE COUNTER
(1)  014442  012704  000001                  MOV     #1,     R4      ;SET UP LINE MARKER
(1)  014446  013700  001346          LOP64:  MOV     TIMER,  R0      ;SET UP TIMER
(1)  014452  050477  164646                  BIS     R4,     @BCSR   ;SET BREAK CONTROL BIT, LINE # IN R1
(1)  014456  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
(1)  014460  001376                          BNE     1$
(1)  014462  017702  164634                  MOV     @RBUF,  R2      ;READ RBUF FOR CHAR PRES
(1)  014466  100403                          BMI     2$              ;BRANCH WHEN FOUND
(1)  014470  005200                          INC     R0              ;WAIT A WHILE
(1)  014472  001371                          BNE     1$
(1)  014474  104002                          HLT+2                   ;CHAR PRES NEVER CAME UP
(1)                                                                 ;R1 = LINE #
(1)                                                                 ;R2 = CONTENTS OF RBUF
(1)
(1)  014476  032702  020000          2$:     BIT     #020000,R2      ;CHECK FOR FRAMING ERROR
(1)  014502  001001                          BNE     .+4             ;BRANCH IF OK
(1)  014504  104002                          HLT+2                   ;FRAMING ERROR NOT UP
(1)                                                                 ;R1 = LINE #
(1)                                                                 ;R2 = CONTENTS OF RBUF
(1)
(1)  014506  010103                          MOV     R1,     R3      ;GET LINE #
(1)  014510  006203                          ASR     R3              ;/4 TO GET RID OF BITS 1 & 0
(1)  014512  006203                          ASR     R3
(1)  014514  006303                          ASL     R3              ;SHIFT BACK FOR EVEN ADDR OF PARITY TABLE
(1)  014516  133763  001366  001300          BITB    DJPAR,PARITY(3) ;CHECK ODD PARITY FLAG
(1)  014524  001404                          BEQ     3$              ;BRANCH IF NOT
(1)  014526  032702  010000                  BIT     #010000,R2      ;CHECK PARITY ERROR
(1)  014532  001005                          BNE     4$              ;
(1)  014534  104002                          HLT+2                   ;ODD PARITY SHOULD CAUSE PARITY ERROR
(1)                                                                 ;R1 = LINE #
(1)                                                                 ;R2 = CONTENTS OF RBUF
(1)
(1)  014536  032702  010000          3$:     BIT     #010000,R2      ;CHECK PARITY ERROR
(1)  014542  001401                          BEQ     4$              ;BRANCH IF OK
(1)  014544  104002                          HLT+2                   ;EVEN PARITY OR NO PARITY
(1)                                                                 ;SHOULDN'T CAUSE PARITY ERROR
(1)                                                                 ;R1 = LINE #
(1)                                                                 ;R2 = CONTENTS OF RBUF
(1)  014546  032702  040000          4$:     BIT     #040000,R2      ;CHECK UART OVERRUN
(1)  014552  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  014554  104002                          HLT+2                   ;UART OVERRUN SET!!
(1)                                                                 ;R1 = LINE #
(1)                                                                 ;R2 = CONTENTS OF RBUF
(1)  014556  010203                          MOV     R2,     R3      ;DUP DATA
(1)  014560  000303                          SWAB    R3              ;LINE # IN LOW BYTE
(1)  014562  042703  177760                  BIC     #177760,R3      ;MASK ALL BUT LINE #
(1)  014566  020103                          CMP     R1,     R3      ;CHECK FOR RIGHT LINE #
(1)  014570  001401                          BEQ     .+4             ;BRANCH IF OK
(1)  014572  104002                          HLT+2                   ;WRONG LINE# IN FI/FO
```

```
(1)                                                         ;R1 = LINE #
(1)                                                         ;R2 = CONTENTS OF RBUF
(1)
(1)
(1)  014574  105702                    TSTB    R2           ;CHECK DATA
(1)  014576  001401                    BEQ     .+4          ;BRANCH IF OK
(1)  014600  104002                    HLT+2                ;WRONG DATA RECEIVED
(1)                                                         ;R1 = LINE #
(1)                                                         ;R2 = CONTENTS OF RBUF
(1)  014602  017702  164514            MOV     @RBUF, R2    ;READ FI/FO AGAIN
(1)  014606  100001                    BPL     .+4          ;BRANCH IF OK
(1)  014610  104002                    HLT+2                ;EXTRA CHAR IN FI/FO
(1)                                                         ;R1 = LINE #
(1)                                                         ;R2 = CONTENTS OF RBUF
(1)  014612  005077  164506            CLR     @BCSR        ;CLEAR BREAK CONTROL REG
(1)  014616  005201                    INC     R1           ;COUNT LINES
(1)  014620  006304                    ASL     R4           ;UPDATE LINE MARKER
(1)  014622  103311                    BCC     LOP64        ;BRANCH IF MORE LINES
(1)
(1)  014624  005077  164470            CLR     @CSR         ;CLEAR CSR
(1)  014630  104400                    SCOPE
(1)
4631 014632  012737  000002  016072    MOV     #2,     TIMES
4632
(1)                         ;*************************************************************************
(1)                         ;TEST 65:       TEST THAT RESET CLEARS ALL BUFFERS
(1)                         ;       NOTE:   THE FI/FO BUFFER IS NOT COMPLETELY CLEARED
(1)                         ;               BY RESET; ONLY CHARACTER PRESENT IS CLEARED.
(1)                         ;PROBABLE FAULTY LOGIC:  M7285 (D2-8) E13
(1)                         ;*************************************************************************
(1)
(1)  014640  052737  000340  177776  TST65:  BIS     #340,@#PS    ;SET PROCESSOR TO LEVEL 7
(1)  014646  012777  177777  164450          MOV     #177777,@TCR ;SET ALL TCR BITS
(1)  014654  012777  052507  164436          MOV     #052507,@CSR ;SET ALL R/W BITS OF CSR
(1)  014662  012777  177777  164434          MOV     #177777,@BCSR ;SET ALL BREAK CONTROL BITS (SEND BREAKS)
(1)                         ;NOTE:  ALL LINES SHOULD BE SENDING BREAKS, BUT NONE SHOULD BERECEIVING
(1)                         ;         BECAUSE THE HALF DUPLEX BIT IS SET
(1)  014670  012777  177777  164430          MOV     #177777,@TBUF ;LOADING TRANS BUFF WHEN BREAK BIT SET
(1)                                                     ;   SHOULD DO NOTHING
(1)  014676  000005                          RESET                ;CLEAR THE WORLD
(1)  014700  013737  001320  014710          MOV     CSR,1$       ;CHECK CSR AND SAVE
(1)  014706  013701                          MOV     @(PC)+,R1
(1)  014710  000000                  1$:     000000
(1)  014712  001401                          BEQ     .+4
(1)  014714  104001                          HLT+1
(1)
(1)  014716  017701  164402                  MOV     @TCR,R1      ;CHECK TCR AND SAVE
(1)  014722  001401                          BEQ     .+4
(1)  014724  104001                          HLT+1
(1)
(1)  014726  017701  164372                  MOV     @BCSR,R1     ;CHECK BCSR AND SAVE
(1)  014732  001401                          BEQ     .+4
(1)  014734  104001                          HLT+1
(1)
(1)  014736  017701  164360                  MOV     @RBUF,R1     ;CHECK RBUF AND SAVE
(1)  014742  100001                          BPL     .+4
```

```
(1)  014744  104001                          HLT+1
(1)
(1)  014746  017701  164354                   MOV     @TBUF,R1        ;CHECK TBUF AND SAVE
(1)  014752  001401                           BEQ     .+4
(1)  014754  104001                           HLT+1
(1)
(1)  014756  104400                           SCOPE
(1)
4633
(1)                                ;**************************************************************************
(1)                                ;TEST 66:       SEND A BINARY COUNT PATTERN ON EACH LINE
(1)                                ;PROBABLE FAULTY LOGIC:  COULD BE ALMOST ANYWHERE!
(1)                                ;**************************************************************************
(1)
(1)  014760  005001                TST66:  CLR     R1              ;SET UP LINE COUNTER
(1)  014762  012703  100000                MOV     #100000,R3      ;SET UP RCV DATA
(1)  014766  012777  015164  164334        MOV     #ISR66,@RCVVEC  ;SET UP RECEIVER INTERRUPT VECTOR
(1)  014774  012777  000240  164330        MOV     #240,   @RCVLVL
(1)  015002  042737  000340  177776        BIC     #340,   @#PS    ;CLEAR PROCESSOR PRIORITY
(1)  015010  052737  000200  177776        BIS     #200,   @#PS    ;SET PRIORITY TO 4
(1)  015016  012700  000001                MOV     #1      R0      ;SET UP LINE MARKER
(1)  015022  012777  010514  164270        MOV     #010514,@CSR    ;BIT2 = MAINTENANCE
(1)                                                                ;BIT3 = CLR MOS
(1)                                                                ;BIT6 = RECEIVER INTERRUPT ENB
(1)                                                                ;BIT8 = TRANS SCAN ENABLE
(1)                                                                ;BIT12= STATUS ENABLE
(1)  015030  032777  000020  164262  10$:    BIT     #BIT4,  @CSR    ;WAIT FOR MOS TO CLEAR
(1)  015036  001374                        BNE     10$
(1)  015040  052777  000001  164252        BIS     #1,@CSR         ;SET THE RCV EN BIT
(1)  015046  005004                1$:     CLR     R4              ;SET FOR OUTPUT DATA
(1)  015050  010077  164250                MOV     R0,     @TCR    ;TRANS CONTROL, ONE LINE AT A TIME
(1)  015054  005777  164240        2$:     TST     @CSR            ;WAIT FOR TRANS READY
(1)  015060  100375                        BPL     2$
(1)  015062  010477  164240                MOV     R4,     @TBUF   ;SEND DATA
(1)  015066  105204                        INCB    R4              ;BINARY COUNT
(1)  015070  123704  001360                CMPB    SUM,R4          ;MAKE SURE THE CORECT NUMBER IS OUTPUTED
(1)  015074  001367                        BNE     2$
(1)  015076  147704  164262        3$:     BICB    @DJLEN,R4       ;SET FOR MAX. RECIEVER COUNT
(1)  015102  120403                        CMPB    R4,R3           ;WAIT FOR RECEIVER DONE
(1)  015104  001374                        BNE     3$
(1)  015106  017702  164206                MOV     @CSR,   R2      ;SAVE CSR
(1)  015112  022702  110505                CMP     #110505,R2      ;CHECK CSR
(1)  015116  001401                        BEQ     .+4             ;BRANCH IF OK
(1)  015120  104002                        HLT+2                   ;CSR ERROR
(1)                                                                ;R1=LINE #
(1)                                                                ;R2=CONTENTS OF CSR
(1)  015122  017702  164174                MOV     @RBUF,  R2      ;CHECK CHARACTER PRESENT
(1)  015126  100001                        BPL     .+4             ;BRANCH IF OK
(1)  015130  104002                        HLT+2                   ;CHARACTER PRESENT SET!!
(1)                                                                ;R1=LINE #
(1)                                                                ;R2=CONTENTS OF CSR
(1)  015132  105003                        CLRB    R3              ;RESET EXPECTED DATA
(1)  015134  062703  000400                ADD     #400,   R3      ;UPDATE LINE # IN EXPECTED DATA
(1)  015140  005201                        INC     R1              ;UPDATE LINE #
(1)  015142  032701  000003                BIT     #3,     R1      ;CHECK FOR FOURTH LINE
(1)  015146  001002                        BNE     4$              ;BRANCH IF NOT
```

```
  (1)  015150  005237  001364          INC     DJLEN           ;MOVE CHARACTER LENGTH POINTER
  (1)  015154  000241            4$:   CLC
  (1)  015156  006300                  ASL     R0              ;UPDATE LINE MARKER
  (1)  015160  103332                  BCC     1$              ;BRANCH IF MORE
  (1)  015162  000416                  BR      END66           ;SKIP ISR
  (1)
  (1)  015164  017702  164132  ISR66:  MOV     @RBUF,  R2      ;READ FIRST DATA
  (1)  015170  100401                  BMI     11$             ;BRANCH IF CHARACTER PRESENT
  (1)  015172  104003                  HLT+3                   ;INTERRUPT BUT NO CHAR PRESENT
  (1)                                                          ;R1=LINE #
  (1)                                                          ;R2=CONTENTS OF RBUF
  (1)                                                          ;R3=EXPECTED DATA
  (1)  015174  147703  164164  11$:    BICB    @DJLEN,R3       ;MASK CHAR. LENGTH
  (1)  015200  020203                  CMP     R2,     R3      ;CHECK THE DATA
  (1)  015202  001401                  BEQ     .+4             ;BRANCH IF OK
  (1)  015204  104003                  HLT+3                   ;DATA ERROR
  (1)                                                          ;R1=LINE #
  (1)                                                          ;R2=CONTENTS OF RBUF
  (1)                                                          ;R3=EXPECTED DATA
  (1)  015206  105203                  INCB    R3              ;UPDATE EXPECTED DATA
  (1)  015210  017702  164106          MOV     @RBUF,  R2      ;READ MORE DATA
  (1)  015214  100767                  BMI     11$             ;BRANCH IF MORE
  (1)  015216  000002            12$:  RTI                     ;RETURN
  (1)
  (1)  015220  162737  000004  001364  END66:  SUB     #4,     DJLEN   ;RESET CHAR LENGTH POINTER
  (1)  015226  013777  001332  164074          MOV     RCVLVL, @RCVVEC ;RESTORE RECEIVER INT. VEC
  (1)  015234  012777  000004  164070          MOV     #IOT,   @RCVLVL
  (1)  015242  005077  164056                  CLR     @TCR            ;CLEAR TCR
  (1)  015246  005077  164046                  CLR     @CSR            ;CLEAR CSR
  (1)  015252  104400                          SCOPE
  (1)
4634   015254  012737  000020  016072          MOV     #20,    TIMES
4635   015262  023737  001362  001344          CMP     DJUUT,UNITS     ;CHECK FOR LAST UNIT
4636   015270  002004                          BGE     DONE            ;BRANCH IF LAST UNIT
4637   015272  005037  001350                  CLR     ALMFLG          ;CLEAR FLAG FOR NEXT UNIT
4638   015276  000137  002330                  JMP     RESTAR          ;JUMP IF NOT
4639
4640   015302                          DONE:
  (1)  015302  004737  020054                  JSR     PC,     KBDINT
  (1)  015306  062737  000001  001316          ADD     #1,PCNT+2       ;ADD 1 TO THE PASS COUNT
  (1)  015314  005537  001314                  ADC     PCNT            ;MAKE IT DOUBLE PREC.
  (1)  015320  000004  017336                  TYPE    ,MEOP           ;END OF PASS INDICATOR
  (1)  015324  032777  002000  164042          BIT     #SW10,@SWR      ;RING THE BELL?
  (1)  015332  001004                          BNE     4$              ;NO!
  (1)  015334  000004  000007                  TYPE    ,BELL           ;RING THE BELL
  (1)  015340  000004  000177                  TYPE    ,177            ;TYPE A FILLER FOR 11/05
  (1)  015344  013700  000042          4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
  (1)  015350  001405                          BEQ     3$              ;IF NONE
  (1)  015352  000005                          RESET                   ;RESET AND
  (1)       015354                             $ENDAD = .
  (1)  015354  004710                          JSR     7,(0)           ;GO TO MONITOR
  (1)  015356  000240                          NOP                     ;SAVE ROOM
  (1)  015360  000240                          NOP                     ;FOR
  (1)  015362  000240                          NOP                     ;ACT11
  (1)  015364  000137  002330          3$:     JMP     RESTAR          ;RETURN
  (1)
```

```
                                ;
                                ;   INITIALIZATION ROUTINE
                                ;
                                ;       DEVICE CSR REGISTER
                                ;
                                ;ON FAILURE:    REGISTER 0 CONTAINS ERROR ADDRESS
                                ;
                                ;SET:
                                ;BIT01 = HALF DUPLEX
                                ;BIT02 = MAINTENANCE
                                ;BIT03 = CLEAR MOS
                                ;BIT06 = RECEIVER INTERRUPT ENABLE
                                ;BIT08 = MASTER XMTR SCAN ENB
                                ;BIT10 = R/W BCSR
                                ;
                                ;WAIT FOR MOS TO CLEAR
                                ;SET:
                                ;BIT00 = RECEIVER ENABLE
                                ;
                                                                SET
015370  012777  002014  163722  INITA:  MOV     #2014,@CSR      ;BIT(S)2,3,10 THEN 0
015376  000413                         BR      INITR
                                ;
015400  012777  000514  163712  INITB:  MOV     #514,@CSR       ;BIT(S)2,3,6,8 THEN 0
015406  000407                         BR      INITR
                                ;
015410  012777  000416  163702  INITC:  MOV     #416,@CSR       ;BIT(S)1,2,3,8 THEN 0
015416  000403                         BR      INITR
                                ;
015420  012777  000414  163672  INITD:  MOV     #414,@CSR       ;BIT(S)2,3,8 THEN 0
                                ;
015426  005000                 INITR:  CLR     R0
                                ;
015430  005200                 1$:     INC     R0              ;ANTIHANG
015432  001004                         BNE     2$              ;ROUTINE
                                ;
015434  011600                         MOV     (SP),R0         ;RECORD SUBROUTINE CALL RETURN
015436  162700  000002                 SUB     #2,R0           ;FORM CALL ADDRESS FOR DISPLAY
                                ;
015442  104000                         HLT                     ;BIT#4 OF DEVICE CSR FAILED TO CLEAR
                                ;
015444  032777  000020  163646  2$:     BIT     #BIT4,@CSR      ;TEST HAS MOS CLEARED
015452  001366                         BNE     1$              ;NO BRANCHES
                                ;                               ;SET:
015454  052777  000001  163636         BIS     #1,     @CSR    ;BIT00 RECEIVE ENABLE
015462  000207                         RTS     PC              ;CONTINUE
                                ;
                                ;
```

```
4694
4695  015464  013701  001320          ALMCK:  MOV     CSR,R1          ;GET CSR ADDR INTO R1
4696  015470  012761  000001  000004          MOV     #1,4(R1)        ;SET LINE 0 IN THE TCR
4697  015476  052711  000004                  BIS     #BIT2, (R1)     ;SET THE MAINT BIT
4698  015502  005037  001356                  CLR     COUNT
4699  015506  005037  001360                  CLR     SUM
4700  015512  005037  001352                  CLR     TIMERA
4701  015516  012737  000200  001354  2$:     MOV     #200,TIMERB     ;SET UP TIME CONSTANTS
4702  015524  005711                          TST     (R1)            ;WAIT FOR TRANSFER READY BIT
4703  015526  100373                          BPL     2$
4704  015530  112761  000377  000006          MOVB    #377,6(R1)      ;OUTPUT A CHAR TO TBUF
4705  015536  005237  001356                  INC     COUNT           ;COUNT EACH CHAR
4706  015542  105711                  1$:     TSTB    (R1)            ;CHECK FOR DONE IN THE CSR
4707  015544  100405                          BMI     3$              ;IF SET GET  OUT OF THE LOOP
4708  015546  004537  015676                  JSR     R5,TIME         ;GIVE DONE TIME TO SET
4709  015552  000773                          BR      1$              ;RETURN TO TEST FOR DONE AGAIN
4710  015554  000760                          BR      2$              ;RETURN TO OUTPUT ANOTHER CHAR
4711  015556  000742                          BR      ALMCK           ;ERROR RETURN TRY AGAIN
4712  015560  042711  000001          3$:     BIC     #BIT0,(R1)      ;TURN OFF RCV ENABLE
4713  015564  022737  000001  001356          CMP     #1,COUNT        ;IF SILO LEVEL SET FOR 1 THEN GET OUT
4714  015572  001414                          BEQ     4$
4715  015574  063737  001356  001360  7$:     ADD     COUNT,SUM       ;CONTINUE TO A COUNT OF 256
4716  015602  023727  001360  000377          CMP     SUM,#377
4717  015610  002771                          BLT     7$
4718  015612  001403                          BEQ     10$             ;IF EQUAL USE IT
4719  015614  163737  001356  001360          SUB     COUNT,SUM       ;IF GREATER SUBTRACT 1 COUNT SO IT'S LESS
4720  015622  000403                  10$:    BR      11$             ;ALL SET GET OUT
4721  015624  012737  000377  001360  4$:     MOV     #377,SUM        ;PUT SUM TO MAX VALUE
4722  015632  004737  020054          11$:    JSR     PC,KBDINT       ;GET THE SWITCH REGISTER
4723  015636  032777  010000  163530          BIT     #SW12,@SWR      ;PRINT ALARM LEVEL?
4724  015644  001413                          BEQ     13$             ;NO
4725  015646  000004  017576                  TYPE,   MALARM
4726  015652  010105                          MOV     R1,TTY          ;YES, PRINT CSR FIRST
4727  015654  004737  016662                  JSR     PC,PRINTR
4728  015660  000004  017516                  TYPE,   MSGDAS
4729  015664  013705  001356                  MOV     COUNT,TTY       ;PRINT ALARM LEVEL
4730  015670  004737  016662                  JSR     PC,PRINTR
4731  015674  000207                  13$:    RTS     PC
4732
4733
4734
4735  015676  105237  001352          TIME:   INCB    TIMERA          ;INCREMENT THROUGH ONE WORD
4736  015702  001012                          BNE     1$              ;GO TEST FOR DONE AGAIN
4737  015704  005337  001354                  DEC     TIMERB          ;MAKE TIMERB LARGER IF FAST PROCESSOR
4738  015710  001007                          BNE     1$
4739  015712  023727  001356  000022          CMP     COUNT,#22       ;HAVE OUTPUTTED 18 TIMES
4740  015720  001002                          BNE     2$              ;NO, GO OUTPUT ANOTHER CHAR
4741  015722  104001                          HLT+1                   ;YES,DONE DID NOT SET AFTER 18 OUTPUTS
4742                                                                  ;R1 = CSR
4743  015724  005725                          TST     (R5)+           ;SET R5 FOR ERROR RETURN
4744  015726  005725                  2$:     TST     (R5)+           ;SET R5 FOR NEXT OUTPUT RETURN
4745  015730  000205                  1$:     RTS     R5              ;RETURN FROM ABOVE OR RETEST DONE
```

```
 4747                                   ;       $SCOPE          SCOPE LOOP HANDLER
  (1)
  (1)                                   ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
  (1)                                   ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
  (1)
  (1)                                   ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
  (1)                                   ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
  (1)
  (1)  015732  004737  020054           TRAP$:  JSR     PC,     KBDINT
  (1)  015736  032777  000400  163430           BIT     #SW8,@SWR       ;LOOP ON SPEC. TEST?
  (1)  015744  001404                           BEQ     1$              ;NO LOOP ON SPEC. TEST
  (1)  015746  127737  163422  001310           CMPB    @SWR,ICNT       ;ON RIGHT TEST?   *SW7-0*
  (1)  015754  001434                           BEQ     OVER$           ;NOT RIGHT TEST
  (1)  015756  032777  040000  163410  1$:      BIT     #SW14,@SWR      ;LOOP ON TEST?
  (1)  015764  001026                           BNE     KIT$            ;LOOP ON TEST IS SET
  (1)  015766  032777  004000  163400           BIT     #SW11,@SWR      ;KILL ITERATIONS
  (1)  015774  001012                           BNE     SVLAD$          ;YES - KILL ITERATIONS
  (1)  015776  105737  001311                   TSTB    ICNT+1          ;FIRST ONE?
  (1)  016002  001404                           BEQ     2$              ;BRANCH IF FIRST
  (1)  016004  123737  016072  001311           CMPB    TIMES,ICNT+1    ;DONE?
  (1)  016012  001013                           BNE     KIT$            ;BRANCH IF NOT
  (1)  016014  112737  000001  001311  2$:      MOVB    #1,ICNT+1       ;FIRST ITERATION
  (1)  016022  105237  001310           SVLAD$: INCB    ICNT            ;COUNT TEST NUMBERS
  (1)  016026  011637  016070                   MOV     (6),LAD         ;SAVE LOOP ADDRESS
  (1)  016032  013737  001310  001376           MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
  (1)  016040  000002                           RTI                     ;RETURN
  (1)
  (1)  016042  105237  001311           KIT$:   INCB    ICNT+1          ;INC THE ITERATION COUNT
  (1)  016046  013737  001310  001376   OVER$:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
  (1)  016054  005737  016070                   TST     LAD             ;FIRST ONE?
  (1)  016060  001760                           BEQ     SVLAD$          ;YES
  (1)  016062  013716  016070                   MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
  (1)  016066  000002                           RTI                     ;FIXES PS
  (1)
  (1)  016070  000000           LAD:    0               ;LOOP ADDRESS
  (1)  016072  000020           TIMES:  20              ;RUN 20 TIMES
```

```
                                ;       $HLT            ERROR TYPEOUT HANDLER
                                ;
                                ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
                                ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
                                ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
                                ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUEMENT
                                ;(HLT+3) WILL BE PLACED IN "HLTCT$:" FOR ADITIONAL TYPEOUTS.
                                ;
016074  004737  020054        EMT$:   JSR     PC,     KBDINT
016100  032777  002000  163266        BIT     #SW10,@SWR      ;BELL ON ERROR?
016106  001402                      BEQ     1$              ;NO - SKIP
016110  000004  000007              TYPE    ,BELL           ;RING BELL
016114  005237  001312        1$:   INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
016120  032777  020000  163246        BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
016126  001026                      BNE     2$              ;SKIP TYPEOUTS
016130  000004  016134              TYPE    .+2             ;.ASCIZ <15><12>
016140  011637  016244              MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
016144  162737  000002  016244      SUB     #2,HLTADR       ;FUDGE ADDRESS
016152  117737  000066  016242      MOVB    @HLTADR,HLTCT$  ;GET HLT ARGUEMENT
016160  013705  016244              MOV     HLTADR,TTY      ;TYPE HLTADR IN OCTAL
016164  004737  016662              JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
016170  000004  016174              TYPE    .+2             ;.ASCIZ " "
016200  004737  016246              JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
016204  005777  163164        2$:   TST     @SWR            ;HALT ON ERROR
016210  100001                      BPL     .+4             ;SKIP IF CONTINUE
016212  000000                      HALT                    ;HALT ON ERROR!
016214  004737  020054              JSR     PC,KBDINT
016220  032777  001000  163146        BIT     #SW9,@SWR       ;CHECK FOR INHIBIT LOOP ON ERROR
016226  001001                      BNE     .+4             ;SKIP IF LOOP ON ERROR
016230  000002                      RTI                     ;RETURN
016232  105037  001311              CLRB    ICNT+1          ;CLEAR ITERATION COUNT
016236  000137  016042              JMP     KIT$            ;LOOP ON TEST UNTIL NO ERRORS
                                ;
016242  000000                HLTCT$: 0                       ;HLT ARGUEMENT
016244  000000                HLTADR: 0                       ;LAST HLT INSTRUCTION EXECUTED

                                ERROR$:
016246  013705  001320              MOV     CSR,TTY         ;TYPE CSR IN OCTAL
016252  004737  016662              JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
016256  042737  007700  016304      BIC     #7700,2$
016264  105337  016242        1$:   DECB    HLTCT$
016270  100411                      BMI     3$
016272  062737  000100  016304      ADD     #100,2$
016300  000004  017347              TYPE,   SPACE
016304  010005                2$:   MOV     %0,TTY          ;TYPE REGISTER X IN OCTAL
016306  004737  016662              JSR     %7,PRINTR
016312  000764                      BR      1$
016314  000207                3$:   RTS     PC
```

```
                                        ;SUBROUTINE TO SAVE INPUT AS OCTAL NUMBER

016316  012737  000001  016610  READIN: MOV     #1,INHRE
016324  004737  016464                  JSR     PC,     READ$   ;GO READ TTY UNTIL CR
016330  005037  016610                  CLR     INHRE
016334  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
(2)     016336  010246                  MOV     R2,-(6)         ;PUSH R2 ON STACK
(2)     016340  010346                  MOV     R3,-(6)         ;PUSH R3 ON STACK
016342  012501                          MOV     (R5)+,  R1
016344  012737  000020  020264          MOV     #20,CNT
016352  012702  016612                  MOV     #INPUT,R2
016356  122712  000120                  CMPB    #120,(R2)       ;CHECK FOR "P"
016362  001425                          BEQ     3$
016364  005011                          CLR     (R1)
016366  112203                  1$:     MOVB    (R2)+,R3
016370  120327  000015                  CMPB    R3,#15
016374  001420                          BEQ     3$
016376  162703  000060                  SUB     #60,R3
016402  032703  177770                  BIT     #177770,R3
016406  001013                          BNE     3$              ;BRANCH IF BAD DATA
016410  006311                          ASL     (R1)
016412  103410                          BCS     2$
016414  006311                          ASL     (R1)
016416  103406                          BCS     2$
016420  006311                          ASL     (R1)
016422  103404                          BCS     2$
016424  050311                          BIS     R3,(R1)
016426  005337  020264                  DEC     CNT
016432  000755                          BR      1$
016434  000244                  2$:     CLZ                     ;MAKE SURE Z-BIT IS CLR
016436  013737  177776  016462  3$:     MOV     @#PS,   PSTEMP  ;SAVE CONDITION CODES
016444  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
(2)     016446  012602                  MOV     (6)+,R2         ;POP STACK INTO R2
(2)     016450  012601                  MOV     (6)+,R1         ;POP STACK INTO R1
016452  013737  016462  177776          MOV     PSTEMP, @#PS    ;RESTORE CONDITION CODES
016460  000205                          RTS     R5

016462  000000          PSTEMP: 0               ;TEMPORARY STORAGE FOR PS

016464  010346          READ$:  MOV     R3,-(6)         ;SAVE R3
(1)     016466  012703  016612  1$:     MOV     #INPUT,R3       ;GET ADDRESS
(1)     016472  022703  016632  2$:     CMP     #INPUT+20,R3    ;BUFFER FULL?
(1)     016476  001415                  BEQ     4$              ;YES - TYPE "?"
(1)     016500  105737  177560          TSTB    @#177560        ;WAIT FOR
(1)     016504  100375                  BPL     .-4             ;A CHARACTER
(1)     016506  113713  177562          MOVB    @#177562,(3)    ;GET CHARACTER
(1)     016512  142713  000200          BICB    #200,(3)        ;GET RID OF JUNK
(1)     016516  122713  000177          CMPB    #177,(3)        ;IS IT A RUBOUT
(1)     016522  001403                  BEQ     4$              ;SKIP IF NOT
(1)     016524  122713  000025          CMPB    #25,(3)
(1)     016530  001006                  BNE     3$
(1)     016532                  4$:
(2)     016532  000004  016536          TYPE    .+2             ;.ASCIZ "?"<15><12>"= "
(1)     016544  000750                  BR      1$              ;ZAP THE BUFFER AND LOOP
(1)     016546  111337  017334  3$:     MOVB    (3),.TYPE       ;SET UP FOR TYPING
```

N06


```
(1)  016552  000004  017334              TYPE    .TYPE           ;ECHO IT
(1)  016556  122723  000015              CMPB    #15,(3)+        ;CHECK FOR RETURN
(1)  016562  001343                      BNE     2$              ;LOOP IF NOT RETURN
(1)  016564  005737  016610              TST     INHRE
(1)  016570  001401                      BEQ     5$
(1)  016572  000402                      BR      6$
(1)  016574  105063  177777      5$:     CLRB    -1(3)           ;ZAP RETURN (THE 15)
(1)  016600  000004  000012      6$:     TYPE    ,12             ;TYPE A LINE FEED
(1)  016604  012603                      MOV     (6)+,R3         ;RESTORE R3
(1)  016606  000207                      RTS     PC              ;RETURN
(1)
(1)  016610  000000              INHRE:  0
(1)  016612  000020              INPUT:  .BLKW   20              ;TTY INPUT AREA
```

```
4805                                    ;       $OCTAL          OCTAL TYPEOUT ROUTINE
 (1)
 (1)                                    ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
 (1)                                    ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
 (1)                                    ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.
 (1)
 (1)  016652  012737  170101  017020  BITYPS: MOV     #170101,.PR     ;SET BIT FLAG ANS 16. CHARACTER COUNT
 (1)  016660  000411                  BR      .PTIT           ;NOW TYPE IT IN BIT FORM
 (1)  016662  112737  000001  017020  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
 (1)  016670  000402                  BR      .+6             ;SKIP
 (1)  016672  005037  017020          PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
 (1)  016676  112737  177772  017021          MOVB    #-6,.PR+1       ;SET COUNT
 (1)  016704  010446                  .PTIT:  MOV     R4,-(6)         ;SAVE R4
 (1)  016706  012704  017022                  MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
 (1)  016712  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
 (1)  016714  000411                          BR      .PRF            ;ROTATE FIRST BIT
 (1)  016716  105014                  .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
 (1)  016720  032737  000100  017020          BIT     #100,.PR        ;BIT TYPING MODE?
 (1)  016726  001004                          BNE     .PRF            ;YES - SKIP 2 ROTATES
 (1)  016730  006105                          ROL     TTY             ;ROTATE BIT INTO C
 (1)  016732  106114                          ROLB    (4)             ;PACK IT
 (1)  016734  006105                          ROL     TTY             ;ROTATE BIT INTO C
 (1)  016736  106114                          ROLB    (4)             ;PACK IT
 (1)  016740  006105                  .PRF:   ROL     TTY             ;ROTATE BIT INTO C
 (1)  016742  106114                          ROLB    (4)             ;PACK IT
 (1)  016744  105714                          TSTB    (4)             ;IS IT ZERO?
 (1)  016746  001402                          BEQ     .+6             ;SKIP INC
 (1)  016750  105237  017020                  INCB    .PR             ;SET FILL SWITCH
 (1)  016754  105737  017020                  TSTB    .PR             ;CHECK FILL SWITCH
 (1)  016760  001402                          BEQ     .+6             ;SKIP BITSET
 (1)  016762  152724  000060                  BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
 (1)  016766  105237  017021                  INCB    .PR+1           ;INC COUNT
 (1)  016772  001351                          BNE     .PRL            ;REPEAT
 (1)  016774  022704  017022                  CMP     #.PR+2,R4       ;EMPTY BUFFER?
 (1)  017000  001002                          BNE     .+6             ;SKIP IF NOT
 (1)  017002  112724  000060                  MOVB    #'0,(4)+        ;LOAD 1 ZERO
 (1)  017006  105014                          CLRB    (4)             ;NULL TERMINATOR
 (1)  017010  000004  017022                  TYPE    .PR+2           ;TYPE IT
 (1)  017014  012604                          MOV     (6)+,R4         ;RESTORE R4
 (1)  017016  000207                          RTS     PC              ;RETURN
 (1)  017020  000012                  .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
 4807  017044  012777  017172  000126  PDOWN$: MOV     #ILLUP,@PUVEC$  ;SET FOR FAST UP
  (1)  017052  012777  000340  000122          MOV     #340,@PUVEC$+2  ;PRIO:7
  (3)  017060  010046                          MOV     R0,-(6)         ;PUSH R0 ON STACK
  (3)  017062  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
  (3)  017064  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
  (3)  017066  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
  (3)  017070  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
  (3)  017072  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
  (1)  017074  010637  017176                  MOV     SP,.SAVR6       ;SAVE SP
  (1)  017100  012777  017110  000072          MOV     #PUP$,@PUVEC$   ;SET UP VECTOR
  (1)  017106  000000                          HALT                    ;WAIT FOR PF
  (1)
  (1)  017110  013706  017176          PUP$:   MOV     .SAVR6,SP       ;GET SP
  (1)  017114  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
  (1)  017116  005201                  1$:     INC     R1              ;WAIT FOR THE INC
  (1)  017120  001376                          BNE     1$              ;OF  WORD
  (3)  017122  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
  (3)  017124  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
  (3)  017126  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
  (3)  017130  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
  (3)  017132  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
  (3)  017134  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
  (1)  017136  012737  017044  000024          MOV     #PDOWN$,@#24    ;SET UP THE POWER DOWN VECTOR
  (1)  017144  012737  000340  000026          MOV     #340,@#26       ;PRIO:7
  (2)  017152  000004  017156                  TYPE    .+2             ;.ASCIZ <15><12>"POWER"
  (1)  017166  000137  002330                  JMP     RESTAR          ;JMP TO USER ADDRESS
  (1)
  (1)  017172  000000          ILLUP:  HALT                            ;THE POWER UP SEQUENCE WAS STARTED
  (1)  017174  000775                  BR      .-2                     ; BEFORE THE POWER DOWN WAS COMPLETE
  (1)
  (1)  017176  000000          .SAVR6: 0                               ;PUT THE SP HERE
  (1)  017200  000024  000026  PUVEC$: 24,26                           ;POWER UP VECTOR
```

```
 4809
  (1)                                ;*****************************************************************************
  (1)                                ;IOT HANDLER.  REENTERENT ROUTINE TO EITHER TYPE MESSAGES OR
  (1)                                ;       INDICATE A FAULSE INTERUPT OR TRAP.
  (1)                                ;*****************************************************************************
  (1)
  (1)  017204  022716  001000        IOTRAP: CMP     #1000,  (SP)    ;CHECK RETURN ADDRESS FOR FAULSE TRAP
  (1)  017210  002407                        BLT     IOT$            ;BRANCH IF "TYPE" COMMAND INTENDED
  (1)  017212  162716  000004                SUB     #4,     (SP)    ;GET VECTOR ADDRESS FROM RETURN ADDRESS
  (1)  017216  012601                        MOV     (SP)+,  R1      ;PUT IN  R1 FOR TYPING
  (1)  017220  005726                        TST     (SP)+           ;POP STACK
  (1)  017222  011602                        MOV     (SP),   R2      ;SAVE RETURN ADDRESS FOR TYPING
  (1)  017224  104002                        HLT+2                   ;UNEXPECTED INTERUPT OR TRAP
  (1)                                                                ;R1 = VECTOR ADDRESS
  (1)                                                                ;R2 = RETURN PC
  (1)  017226  000002                        RTI                     ;CONTINUE THE PROGRAM
  (1)
  (1)                                ;MCALL  $TYPE
  (2)                                ;       $TYPE           MESSAGE TYPEOUT ROUTINE
  (2)
  (2)                                ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
  (2)                                ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
  (2)                                ;MESSAGE STARTING IN LOCATION "ADR:"  2) "TYPE ,CHAR" - TYPES
  (2)                                ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
  (2)                                ;THE MESSAGE WHICH IS INLINE ASCII.
  (2)
  (2)  017230  010546                IOT$:   MOV     TTY,-(6)        ;SAVE TTY
  (2)  017232  017605  000002                MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
  (2)  017236  032705  177400                BIT     #177400,TTY     ;IS IT A TYPEM?
  (2)  017242  001004                        BNE     1$              ;NO
  (2)  017244  010537  017334                MOV     TTY,.TYPE       ;GET THE CHARACTER
  (2)  017250  012705  017334                MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
  (2)  017254  105715                1$:     TSTB    (TTY)           ;TERMINATOR?
  (2)  017256  001406                        BEQ     2$              ;GET OUT IF SO
  (2)  017260  112537  177566                MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
  (2)  017264  105737  177564                TSTB    @#177564        ;IS THE PRINTER READY
  (2)  017270  100375                        BPL     .-4             ;WAIT UNTIL IT IS
  (2)  017272  000770                        BR      1$              ;GET THE NEXT CHARACTER
  (2)  017274  017646  000002        2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
  (2)  017300  062766  000002  000004        ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
  (2)  017306  022666  000002                CMP     (6)+,2(6)       ;IS IT .+2?
  (2)  017312  001006                        BNE     3$              ;NO
  (2)  017314  062705  000002                ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
  (2)  017320  042705  000001                BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
  (2)  017324  010566  000002                MOV     TTY,2(6)        ;RESTORE ADDRESS
  (2)  017330  012605                3$:     MOV     (6)+,TTY        ;RESTORE TTY
  (2)  017332  000002                        RTI                     ;RETURN
  (2)  017334  000000                .TYPE:  0                       ;CHARACTER TYPE LOCATION
 4810
 4811  017336  005015  047505  000120  MEOP:   .ASCIZ <15><12> "EOP"
 4812  017344  005015     000          RETURN: .ASCIZ  <15><12>
 4813  017347     040  000040          SPACE:  .ASCIZ  " "
 4814  017352  005015  044506  051522  MSGADR: .ASCIZ  <15><12>"FIRST DJ11 ADDRESS:  "
       017360  020124  045104  030461
       017366  040440  042104  042522
       017374  051523  020072  000040
 4815  017402  005015  042526  052103  MSGVEC: .ASCIZ  <15><12>"VECTOR ADDRESS:  "
```

```
                017410  051117  040440  042104
                017416  042522  051523  020072
                017424  000040
 4816   017426  005015  047516  020056  MSGNUM: .ASCIZ  <15><12>"NO. OF DJ11'S:  "
                017434  043117  042040  030512
                017442  023461  035123  020040
                017450     000
 4817   017451     015  051412  040524  MSGCON: .ASCIZ  <15><12>"STANDARD CONFIGURATION?  "
                017456  042116  051101  020104
                017464  047503  043116  043511
                017472  051125  052101  047511
                017500  037516  020040     000
 4818   017505     015  046012  047111  MSGLIN: .ASCIZ  <15><12>"LINES "
                017512  051505  000040
 4819   017516  026440  000040          MSGDAS: .ASCIZ  " - "
 4820   017522  005015  044103  051101  MSGLEN: .ASCIZ  <15><12>"CHAR LENGTH:  "
                017530  046040  047105  052107
                017536  035110  020040     000
 4821   017543     015  050012  051101  MSGPAR: .ASCIZ  <15><12>"PARITY(NO, ODD, EVEN):  "
                017550  052111  024131  047516
                017556  020054  042117  026104
                017564  042440  042526  024516
                017572  020072  000040
 4822   017576  005015  044523  047514  MALARM: .ASCIZ <15><12>"SILO ALARM LEVEL FOR CSR"<15><12>
                017604  040440  040514  046522
                017612  046040  053105  046105
                017620  043040  051117  041440
                017626  051123  005015     000
 4823   017633     015  046412  044501  MTITLE: .ASCIZ <15><12> "MAINDEC-11-DZDJA-E  DJ11 LOGIC TESTS"<15><12>
                017640  042116  041505  030455
                017646  026461  055104  045104
                017654  026501  020105  042040
                017662  030512  020061  047514
                017670  044507  020103  042524
                017676  052123  006523  000012
 4824                                   .EVEN
 4828
 4829                                   ;SUBROUTINE TO AUTOMATICLY DETERMINE THE NUMBER OF DJ11'S ON THE SYSTEM
 4830                                   ;       AND WHERE THEIR INTERUPT VECTORS ARE.
 4831                                   ;THIS ROUTINE IS ONLY USED IF LOC 42 IS NOT ZERO, AS WHEN THE PROGRAM IS
 4832                                   ;       BEING RUN UNDER ACT11 OR DDP MONITOR CONTROL.
 4833                                   ;NOTE:  SOME OF THE LOGIC MUST BE FUNCTIONAL OR THIS ROUTINE WILL BOMB!
 4834
 4835   017704  012700  160000          AUTO:   MOV     #160000,R0      ;START AT NO RESPONSE BASE DJ11 -10(8)
 4836   017710  012702  000001                  MOV     #1,     R2      ;COUNTER OF NON RESPONSE OR DJ11 OR DONE
 4837   017714  012737  000002  000006          MOV     #RTI,   @#6     ;RTI WHEN TIME-OUT
 4838   017722  005001                  5$:     CLR     R1              ;SET UP COUNTER
 4839   017724  000261                  1$:     SEC                     ;SET CARRY
 4840   017726  005710                          TST     (R0)            ;CHECK FOR ANY DJ11'S
 4841   017730  103404                          BCS     2$              ;BRANCH IF IT TIMED OUT
 4842   017732  062700  000010                  ADD     #10,    R0      ;POINT TO NEXT DJ11 ADDRESS
 4843   017736  005201                          INC     R1              ;COUNT DJ11'S
 4844   017740  000771                          BR      1$              ;LOOK FOR MORE
 4845
 4846   017742  005302                  2$:     DEC     R2              ;COUNT DOWN DEVICES
 4847   017744  100405                          BMI     7$              ;BRANCH IF DONE
```

```
 4848  017746  062700  000010                ADD     #10,    R0      ;POINT TO FIRST DJ11
 4849  017752  010037  001340                MOV     R0,     DEVADR  ;SAVE FIRST DJ11 ADDRESS
 4850  017756  000761                        BR      5$              ;GO COUNT DJ11'S
 4851
 4852  017760  005037  000006          7$:   CLR     @#6             ;RESTORE TIME-OUT CATCHER
 4853  017764  010137  001344                MOV     R1,     UNITS   ;SAVE COUNT
 4854  017770  001003                        BNE     3$              ;BRANCH IF NOT ZERO
 4855  017772  104000                        HLT                     ;REPORT THAT NO DJ11'S WERE FOUND
 4856  017774  000137  015302                JMP     DONE            ;EXIT THIS PROGRAM
 4857
 4858                                  ;ROUTINE TO DETERMINE VECTOR ADDRESSES
 4859
 4860  020000  013746  000020          3$:   MOV     @#20,   -(SP)   ;SAVE IOT VECTOR ON THE STACK
 4861  020004  012737  020040  000020        MOV     #4$,    @#20    ;RESET IOT VECTOR
 4862  020012  013701  001340                MOV     DEVADR, R1      ;GET FIRST DJ ADR
 4863  020016  012721  040400                MOV     #40400, (R1)+   ;SET CSR
                                                                      ;BIT8 = TRANS SCAN ENABLE
 4864                                                                 ;BIT14 = TRANS INTERUPT ENABLE
 4865
 4866  020022  005721                        TST     (R1)+           ;INC POINTER
 4867  020024  012721  000001                MOV     #1,     (R1)+   ;SET TCR, LINE 0
 4868  020030  000001                        WAIT                    ;WAIT FOR AN INTERUPT
 4869  020032  012637  000020                MOV     (SP)+,  @#20    ;RESTORE IOT VECTOR
 4870  020036  000207                        RTS     PC
 4871
 4872  020040  162716  000010          4$:   SUB     #10,    (SP)    ;REPOSITION ADDRESS TO RVC VEC
 4873  020044  011637  001342                MOV     (SP),   VECADR  ;SAVE FIRST VECTOR
 4874  020050  022626                        CMP     (SP)+,  (SP)+   ;RESET STACK FROM IOT
 4875  020052  000002                        RTI                     ;RETURN FROM INTERUPT - RESTORE STATUS
 4876
 4877
  (1)  020054  022737  000176  001374  KBDINT: CMP     #SWREG,SWR
  (1)  020062  001016                        BNE     1$
  (1)  020064  005037  020122                CLR     TMP1
  (1)  020070  113737  177562  020122        MOVB    177562,TMP1
  (1)  020076  142737  000200  020122        BICB    #200,TMP1
  (1)  020104  122737  000007  020122        CMPB    #7,TMP1
  (1)  020112  001002                        BNE     1$
  (1)  020114  004737  020204                JSR     PC,CNTLU
  (1)  020120  000207                  1$:   RTS     PC
  (1)
  (1)  020122  000000                  TMP1: 0
  (1)
 4878
  (1)  020124  013746  000006          SUSWRR: MOV     6,-(SP)
  (1)  020130  013746  000004                  MOV     4,-(SP)
  (1)  020134  012737  020154  000004          MOV     #1$,4
  (1)  020142  022777  177777  161224          CMP     #-1,@SWR
  (1)  020150  001402                          BEQ     2$
  (1)  020152  000407                          BR      3$
  (1)  020154  022626                    1$:   CMP     (SP)+,(SP)+
  (1)  020156  012737  000176  001374    2$:   MOV     #SWREG,SWR
  (1)  020164  012737  000174  001376          MOV     #DISPREG,DISPLAY
  (1)  020172  012637  000004          3$:   MOV     (SP)+,4
  (1)  020176  012637  000006                MOV     (SP)+,6
  (1)  020202  000207                        RTS     PC
  (1)
```

```
 4879
  (1)  020204  022737  000176  001374  CNTLU:  CMP     #SWREG,SWR
  (1)  020212  001023                          BNE     1$
  (1)  020214  000004  020276                  TYPE    ,SWREQ
  (2)  020220  013705  000176                  MOV     SWREG,TTY       ;TYPE SWREG IN OCTAL
  (2)  020224  004737  016662                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
  (1)  020230  000004  020266                  TYPE    ,NEWIS
  (1)  020234  004537  016316                  JSR     R5,READIN
  (1)  020240  020122                          .WORD   TMP1
  (1)  020242  001360                          BNE     CNTLU
  (1)  020244  022737  000020  020264          CMP     #20,CNT
  (1)  020252  001403                          BEQ     1$
  (1)  020254  013777  020122  161112          MOV     TMP1,@SWR
  (1)  020262  000207                  1$:     RTS     PC
  (1)
  (1)  020264  000000                  CNT:    0
  (1)
  (1)  020266  020040  042516  036527  NEWIS:  .ASCIZ  "  NEW= "
  (1)  020274  000040
  (1)  020276  005015  053523  036522  SWREQ:  .ASCIZ  <15><12>"SWR= "
  (1)  020304  000040
  (1)
 4880          000001                  .END
```

```
ALMCK   015464          4622    4695#   4711
ALMFLG  001350          2668#   2703*   4622*   4637*
AUTO    017704          2706    4835#
BCSR    001324          2656#   4628*   4629*   4630*   4632*
BEGIN   001402          2625    2689#
BELL  = 000007          2619#   4640    4750
BITYPS  016652          4805#
BIT0  = 000001          2619#   4568    4576    4578    4579    4580    4581    4582    4583    4584    4585    4586    4589
                        4620    4621    4622    4623    4624    4625    4627    4712
BIT1  = 000002          2619#   4561    4590    4623
BIT10 = 002000          2619#   4565    4603
BIT11 = 004000          2619#   4604
BIT12 = 010000          2619#   4566    4607    4626
BIT13 = 020000          2619#   4608    4626
BIT14 = 040000          2619#   4567    4609    4627
BIT15 = 100000          2619#   4610
BIT2  = 000004          2619#   4562    4568    4591    4697
BIT3  = 000010          2619#   4568    4592    4620    4626
BIT4  = 000020          2619#   4568    4595    4620    4622    4627    4633    4683
BIT5  = 000040          2619#   4596
BIT6  = 000100          2619#   4563    4597
BIT7  = 000200          2619#   4598    4621
BIT8  = 000400          2619#   4564    4601
BIT9  = 001000          2619#   4602
CLRVEC  002364          2840    2842    2845#
CNT     020264          4775*   4793*   4879#
CNTLU   020204          2843    4877    4879#
CONFIG  002004          2741    2772#   2787
COUNT   001356          2671#   4622    4623    4624    4625    4626    4698*   4705*   4713    4715    4719    4729    4739
CSR     001320          2653#   2851*   2858*   2863    4560*   4561*   4562*   4563*   4564*   4565*   4566*   4567*   4568*
                        4569*   4570*   4576*   4577*   4578*   4579*   4580*   4581*   4582*   4583*   4584*   4585*   4586*
                        4589*   4590*   4591*   4592*   4595*   4596*   4597*   4598*   4601*   4602*   4603*   4604*   4607*
                        4608*   4609*   4610*   4620*   4621*   4622*   4623*   4624*   4625*   4626*   4627*   4628*   4630*
                        4632*   4633*   4662*   4665*   4668*   4671*   4683    4686*   4695    4753
DEFADR  001370          2677#   2717
DEFVEC  001372          2678#
DEVADR  001340          2664#   2713    2715    2717*   2718*   2719    2858    4849*   4862
DISPLA  001376          2679#   4747*   4878*
DISPRE  000174          2625#   4878
DJLEN   001364          2675#   2853*   2860*   4589    4590    4591    4592    4593*   4595    4596    4597    4598    4599*
                        4601    4602    4603    4604    4605*   4607    4608    4609    4610    4611*   4633*
DJMXNO= 000020          2622#   2767
DJPAR   001366          2676#   2854*   2861*   4629    4630
DJUUT   001362          2674#   2770*   2855    2857*   2862*   4635
DONE    015302          4636    4640#   4856
EMT$    016074          2696    4750#
END22   004462          4579#
END23   004602          4580#
END24   004722          4581#
END25   005042          4582#
END26   005162          4583#
END27   005302          4584#
END30   005422          4585#
END31   005542          4586#
END56   012524          4624#
END57   012706          4625#
```

```
END66   015220          4633#
ERRORS  001312          2650#   2700*   4750*
ERROR$  016246          4750    4752#
FTIME   001400          2681#   2839    2844*
GETADR  001514          2705    2709#   2714    2720
GETLEN  002142          2799#   2805    2809    2813
GETNUM  001650          2734#   2743    2745    2750    2752    2755    2768
GETPAR  002226          2803    2815#   2825
GETVEC  001572          2722#   2725    2731    2733
HLT   = 104000          2619#   4560    4561    4562    4563    4564    4565    4566    4567    4568    4569    4570    4571
                        4572    4573    4576    4577    4578    4579    4580    4581    4582    4583    4584    4585    4586
                        4589    4590    4591    4592    4595    4596    4597    4598    4601    4602    4603    4604    4607
                        4608    4609    4610    4620    4621    4622    4623    4624    4625    4626    4627    4628    4629
                        4630    4632    4633    4681    4741    4809    4855
HLTADR  016244          4750#*
HLTCT$  016242          4750#*  4755*
ICNT    001310          2649#   2878*   4747*   4750*
ILLUP   017172          4807#
INHRE   016610          2800*   2816*   4770*   4772*   4803#
INITA   015370          4629    4630    4662#
INITB   015400          4624    4625    4665#
INITC   015410          4623    4668#
INITD   015420          4589    4590    4591    4592    4595    4596    4597    4598    4601    4602    4603    4604    4607
                        4608    4609    4610    4620    4621    4622    4626    4671#
INITR   015426          4663    4666    4669    4673#
INPUT   016612          2737    2774    2782    2784    2786    2802    2804    2806*   2810*   2818    2820    2822    2824
                        4776    4803#
IOTRAP  017204          2692    4809#
IOT$    017230          4809#
ISR22   004452          4579#
ISR23   004572          4580#
ISR24   004712          4581#
ISR25   005034          4582#
ISR26   005154          4583#
ISR27   005274          4584#
ISR30   005414          4585#
ISR31   005534          4586#
ISR56   012514          4624#
ISR57   012700          4625#
ISR66   015164          4633#
KBDINT  020054          4640    4722    4747    4750    4877#
KIT$    016042          4747#   4750
LAD     016070          2879*   4575*   4588*   4594*   4600*   4606*   4619*   4747#*
LENGTH  001200          2644#   2777    2790    2860
LEVEL7= 000340          2620#   4560
LOP32   005622          4589#
LOP33   006012          4590#
LOP34   006202          4591#
LOP35   006372          4592#
LOP36   006574          4595#
LOP37   006764          4596#
LOP40   007154          4597#
LOP41   007344          4598#
LOP42   007546          4601#
LOP43   007736          4602#
LOP44   010126          4603#
```

```
LOP45   010316          4604#
LOP46   010520          4607#
LOP47   010710          4608#
LOP50   011100          4609#
LOP51   011270          4610#
LOP61   013774          4627#
LOP64   014446          4630#
MALARM  017576          4725    4822#
MEOP    017336          4640    4811#
MSGADR  017352          2711    4814#
MSGCON  017451          2772    4817#
MSGDAS  017516          2795    4728    4819#
MSGLEN  017522          2799    4820#
MSGLIN  017505          2793    4818#
MSGNUM  017426          2734    4816#
MSGPAR  017543          2815    4821#
MSGVEC  017402          2722    4815#
MTITLE  017633          2709    4823#
N     = 000067          2561#   4560#   4561#   4562#   4563#   4564#   4565#   4566#   4567#   4568#   4569#   4570#   4571#
                        4572#   4573#   4576#   4577#   4578#   4579#   4580#   4581#   4582#   4583#   4584#   4585#   4586#
                        4589#   4590#   4591#   4592#   4595#   4596#   4597#   4598#   4601#   4602#   4603#   4604#   4607#
                        4608#   4609#   4610#   4620#   4621#   4622#   4623#   4624#   4625#   4626#   4627#   4628#   4629#
                        4630#   4632#   4633#
NEWIS   020266          4879#
OPEN  = 000000          2621#
OVERS   016046          4747#
PARITY  001300          2645#   2791    2830    4629    4630
PARIT2  001302          2646#
PARIT3  001304          2647#
PARIT4  001306          2648#
PCNT    001314          2651#   2701*   2702*   4640*
PDOWNS  017044          2694    4807#
PRINTR  016662          4727    4730    4750    4753    4760    4805#   4879
PRINTS  016672          2794    2797    4805#
PS    = 177776          2619#   4579*   4580*   4581*   4582*   4583*   4584*   4585*   4586*   4624*   4625*   4626*   4632*
                        4633*   4796    4798*
PSTEMP  016462          4796*   4798    4801#
PUPS    017110          4807#
PUVECS  017200          4807#*
RBUF    001322          2654#   2865*   4589    4590    4591    4592    4595    4596    4597    4598    4601    4602    4603
                        4604    4607    4608    4609    4610    4620    4621    4622    4623    4624    4625    4626    4627
                        4629    4630    4632    4633
RCVLVL  001332          2660#   2873*   4624*   4625*   4626*   4633#
RCVVEC  001330          2659#   2852*   2859*   2871    4624*   4625*   4626*   4633*
READIN  016316          2712    2723    4770#   4879
READS   016464          2735    2773    2801    2817    4771    4803#
RESTAR  002330          2627    2707    2775    2783    2785    2837#   4638    4640    4807
RETURN  017344          2710    4812#
SCOPE = 104400          2619#   2880    4560    4561    4562    4563    4564    4565    4566    4567    4568    4569    4570
                        4571    4572    4573    4576    4577    4578    4579    4580    4581    4582    4583    4584    4585
                        4586    4589    4590    4591    4592    4595    4596    4597    4598    4601    4602    4603    4604
                        4607    4608    4609    4610    4620    4621    4622    4623    4624    4625    4626    4627    4628
                        4629    4630    4632    4633
SPACE   017347          4758    4813#
STACK = 001200          2623#   2689    2838
SUM     001360          2672#   4633    4699*   4715*   4716    4719*   4721*
```

K07


```
SUSWRR  020124          2690    4878*
SVLAD$  016022          4747*
SWR     001374          2679*   2841    4640    4723    4747    4750    4877    4878*   4879*
SWREG   000176          2625*   2841    4877    4878    4879
SWREQ   020276          4879*
SW10  = 002000          2571*   4640    4750
SW11  = 004000          2571*   4747
SW12  = 010000          2571*   4723
SW13  = 020000          2571*   4750
SW14  = 040000          2571*   4747
SW15  = 100000          2571*
SW8   = 000400          2571*   4747
SW9   = 001000          2571*   4750
TBUF    001326          2657*   2870*   4576    4577    4589*   4590*   4591*   4592*   4595*   4596*   4597*   4598*   4601*
                        4602*   4603*   4604*   4607*   4608*   4609*   4610*   4620*   4621*   4622*   4623*   4624*   4625*
                        4626*   4627*   4632*   4633*
TCR     001324          2655*   2867*   4571*   4572*   4573*   4576*   4577*   4578*   4579*   4580*   4581*   4582*   4583*
                        4584*   4585*   4586*   4589*   4590*   4591*   4592*   4595*   4596*   4597*   4598*   4601*   4602*
                        4603*   4604*   4607*   4608*   4609*   4610*   4620*   4621*   4622*   4623*   4624*   4625*   4626*
                        4627*   4632*   4633*
TIME    015676          4708    4735*
TIMER   001346          2667*   4574*   4576*   4577*   4578    4587*   4589*   4590*   4591*   4592*   4595*   4596*   4597*
                        4598*   4601*   4602*   4603*   4604*   4607*   4608*   4609*   4610*   4612*   4613    4617*   4618*
                        4620    4621    4622    4623    4626    4627    4629    4630
TIMERA  001352          2669*   4700*   4735*
TIMERB  001354          2670*   4701*   4737*
TIMES   016072          4631*   4634*   4747*
TMP1    020122          4877**  4879
TRAP$   015732          2698    4747*
TST1    002602          4560*
TST10   003256          4567*
TST11   003330          4568*
TST12   003472          4569*
TST13   003560          4570*
TST14   003634          4571*
TST15   003674          4572*
TST16   003762          4573*
TST17   004052          4576*
TST2    002662          4561*
TST20   004162          4577*
TST21   004276          4578*
TST22   004376          4579*
TST23   004516          4580*
TST24   004636          4581*
TST25   004756          4582*
TST26   005076          4583*
TST27   005216          4584*
TST3    002734          4562*
TST30   005336          4585*
TST31   005456          4586*
TST32   005610          4589*
TST33   006000          4590*
TST34   006170          4591*
TST35   006360          4592*
TST36   006562          4595*
TST37   006752          4596*
```

```
TST4     003006          4563#
TST40    007142          4597#
TST41    007332          4598#
TST42    007534          4601#
TST43    007724          4602#
TST44    010114          4603#
TST45    010304          4604#
TST46    010506          4607#
TST47    010676          4608#
TST5     003060          4564#
TST50    011066          4609#
TST51    011256          4610#
TST52    011510          4620#
TST53    011674          4621#
TST54    012102          4622#
TST55    012264          4623#
TST56    012416          4624#
TST57    012600          4625#
TST6     003132          4565#
TST60    012762          4626#
TST61    013746          4627#
TST62    014220          4628#
TST63    014266          4629#
TST64    014434          4630#
TST65    014640          4632#
TST66    014760          4633#
TST7     003204          4566#
TYPE   = 000004          2619#   2709    2710    2711    2722    2734    2772    2793    2795    2799    2815    4640    4725
                         4728    4750    4758    4803    4805    4807    4879
TYPLIN   002110          2793#   2829    2834
T60A     013076          4626#
T60B     013142          4626#
T60C     013232          4626#
T60D     013446          4626#
UNITS    001344          2666#   2769*   2833    2855    4635    4853*
VECADR   001342          2665#   2724    2726    2728*   2729*   2730    2732    2859    4873*
XMTLVL   001336          2662#   2877*   4579*   4580*   4581*   4582*   4583*   4584*   4585*   4586*
XMTVEC   001334          2661#   2875*   4579*   4580*   4581*   4582*   4583*   4584*   4585*   4586*
$ENDAD=  015354          2625    4640#
.      = 020306          2625#   2626#   2632#   2644#   4561    4562    4563    4564    4565    4566    4567    4568    4569
                         4570    4571    4572    4573    4575    4576    4577    4578    4588    4589    4590    4591    4592
                         4594    4595    4596    4597    4598    4600    4601    4602    4603    4604    4606    4607    4608
                         4609    4610    4619    4620    4621    4622    4623    4624    4625    4626    4627    4628    4629
                         4630    4632    4633    4640    4747    4750#   4803#   4805#   4807    4809
.BIT   = 177777          2560#   2571    4640    4747    4750
.PR      017020          4805#*
.PRF     016740          4805#
.PRL     016716          4805#
.PTIT    016704          4805#
.SAVR6   017176          4807#*
.TYPE    017334          4803*   4809#*
```

```
BITYPE     525#   2619#
BLKBLK    1196#
BUFHDR    1189#
DUMP       515#   2619#   4750    4753    4879
DUMP18     538#
FILBLK    1181#
LNKBLK    1173#
ODT11     1224#
ODT11X    1740#
POP        571#   2619#   4797    4807
PRINT      530#   2619#   4750    4803    4807
PUSH       564#   2619#   4773    4807
SCOPE.     353#
SCOP.      382#
SDUMP      520#   2619#   2794    2797
TRACE     1154#
TRNBLK    1202#
TYPEM      543#
WRITE     1210#
$BRK      4376#   4630
$BRKO     4303#   4629
$BYTRF    3030#   4569    4572
$CATCH     578#
$CLREG    3062#   4570    4573
$CMOS     3126#   4620
$CMTAG     293#
$CNTL     2475#   4879
$CSREF    2885#   4560
$DONE     3517#   4589    4590    4591    4592    4595    4596    4597    4598    4601    4602    4603    4604    4607    4608
          4609    4610
$END       606#   4640
$EQUAT     451#   2619
$EXER     4471#   4633
$FIFOV    3811#   4626
$HD       3752#   4623
$HLT       725#   4750
$IOCAT    2508#   2625
$IOT      2536#   4809
$KBIN     2459#   4877
$KRAT      134#
$LOADR     397#
$LVL      3359#   4579    4580    4581    4582    4583    4584    4585    4586
$MTSE     3475#   4578
$OCTAL     789#   4805
$POWER     871#   4807
$RAND     1074#
$RAND4    1117#
$RCVDA    3684#   4622
$RCVE     2947#   4568
$READ     1023#   4803
$REGBC    2918#   4561    4562    4563    4564    4565    4566    4567
$REGST    3095#   4571    4628
$RESET    3204#   4632
$RINT     3406#   4624    4625
$RONLY    3003#
$RUOR     4075#
```

```
$SCOPE     654#   4747
$SETUP     311#
$SRAT      191#
$SWDOC     103#   2571
$SWRDF    2497#   2679
$SWRRR    2442#   4878
$TRAP      978#
$TYPE      919#   4809
$URAT      242#
$XRDY     3300#   4577
$XRDYC    3604#   4621
$XRDYO    3253#   4576
$XUOR     4211#   4627
.SCOP      374#
.SCOPE     345#


. ABS.  020306      000

 ERRORS DETECTED:  0

 DZDJAE,DZDJAE/CRF=DZDJAE.MAC,DZDJAE.P11
 RUN-TIME: 9 13 .6 SECONDS
 RUN-TIME RATIO: 253/24=10.3
 CORE USED:  32K  (63 PAGES)
```