DJ11

LOGIC TEST

MD-11-DZDJA-D

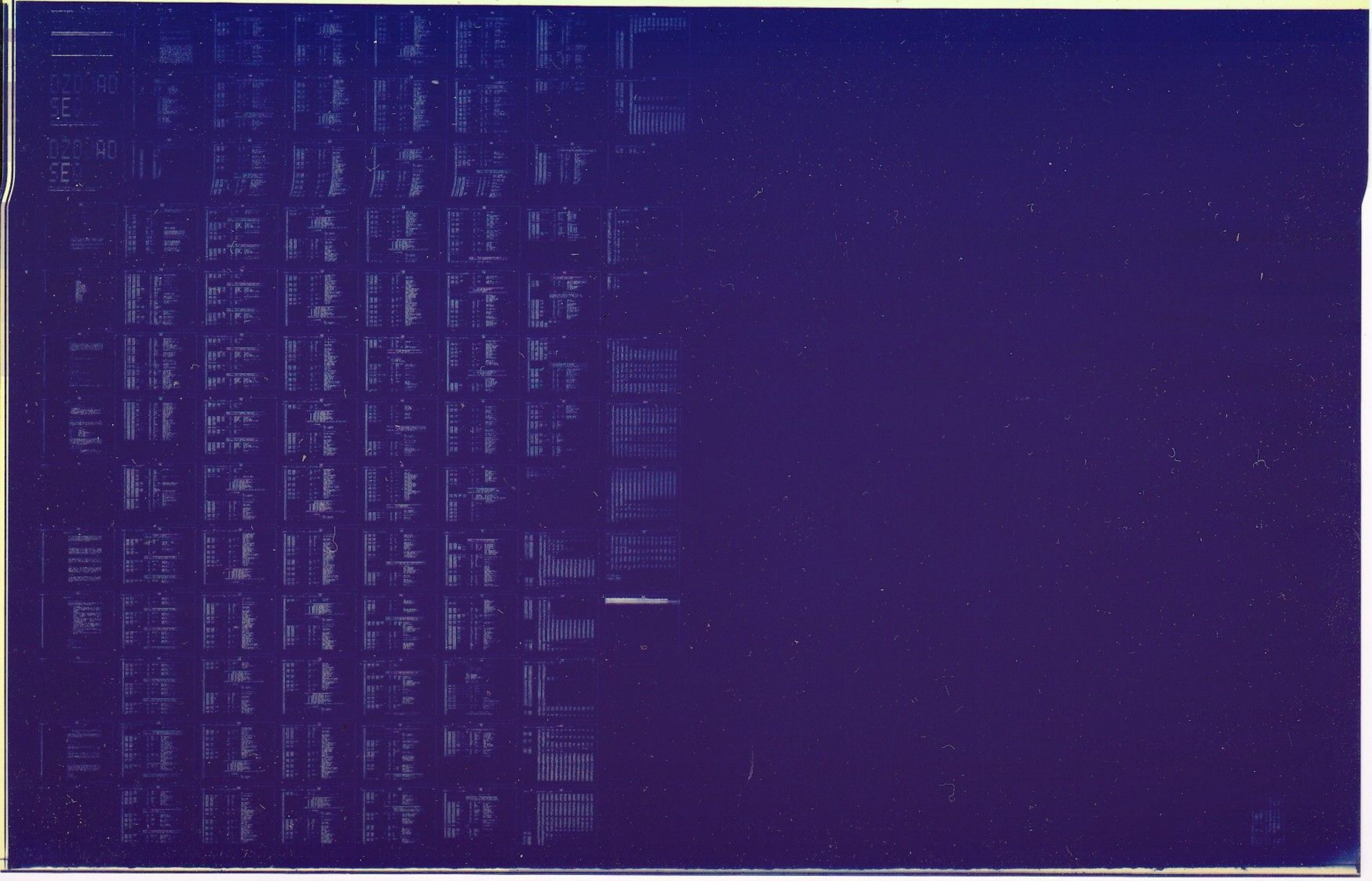
EP-DZDJA-D-DL-A

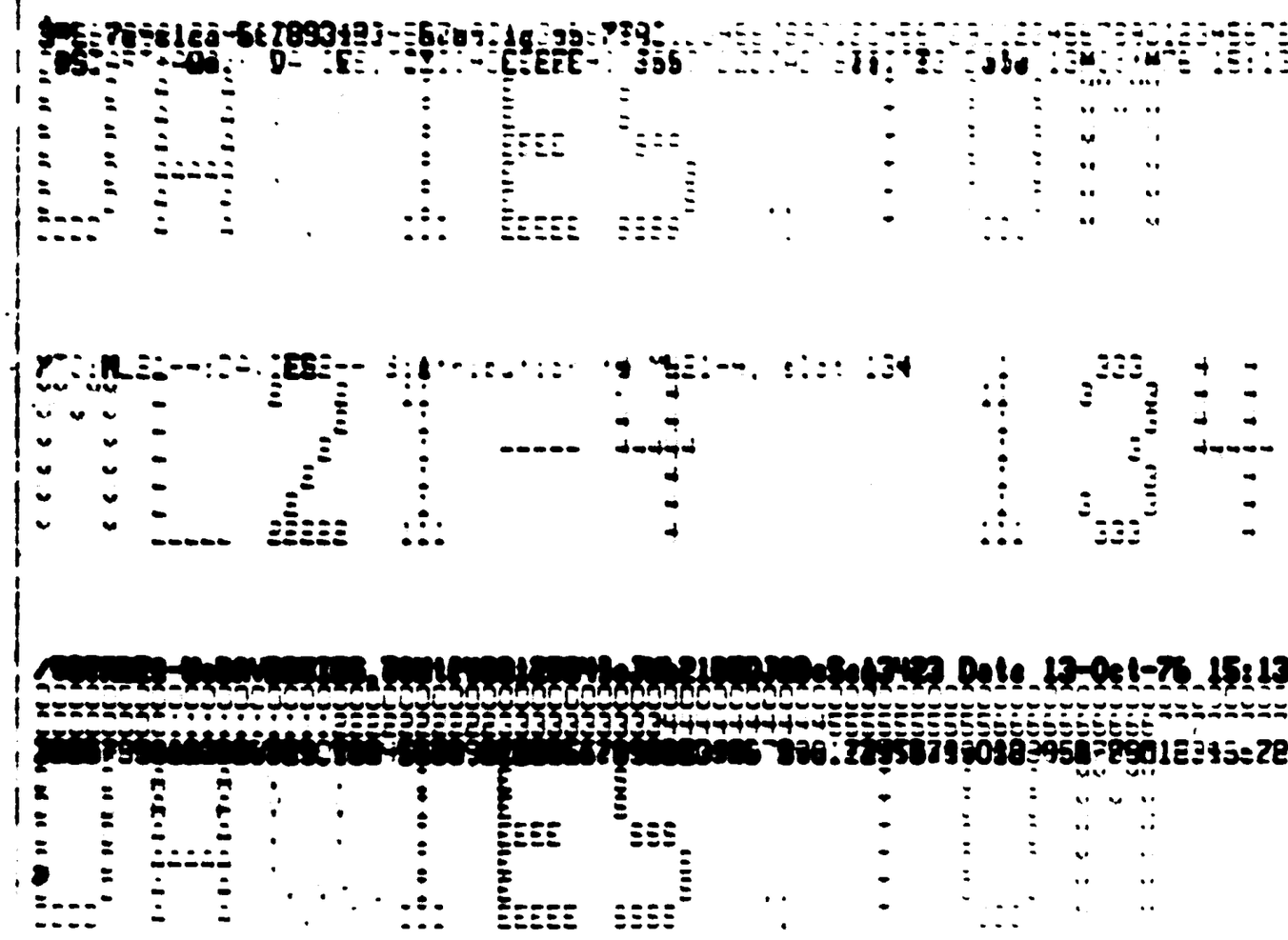
FICHE 1 OF 1

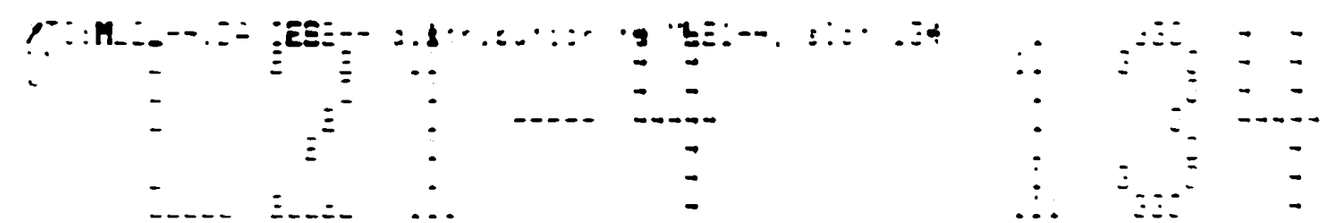
NOV 1976

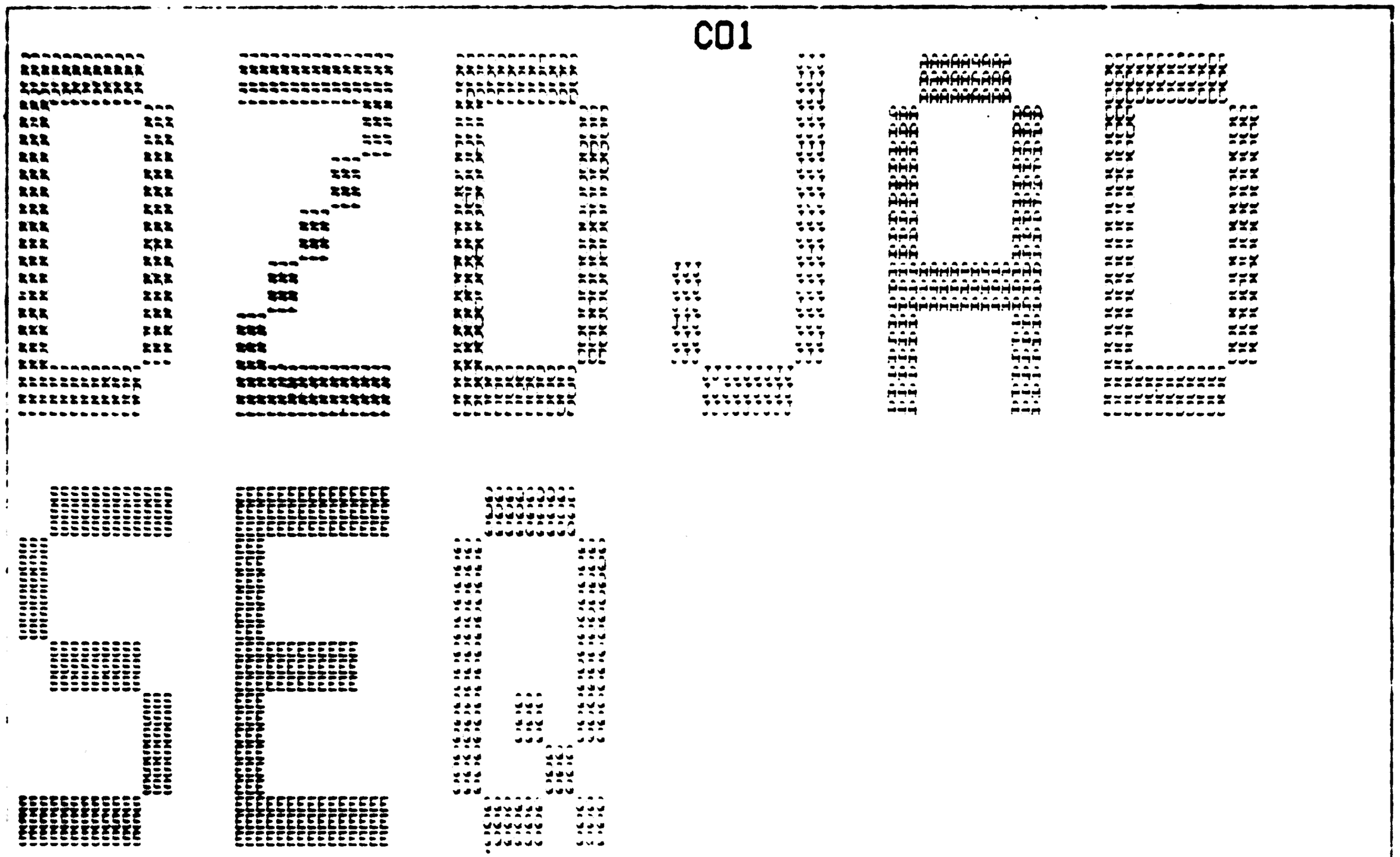
digital

MADE IN USA

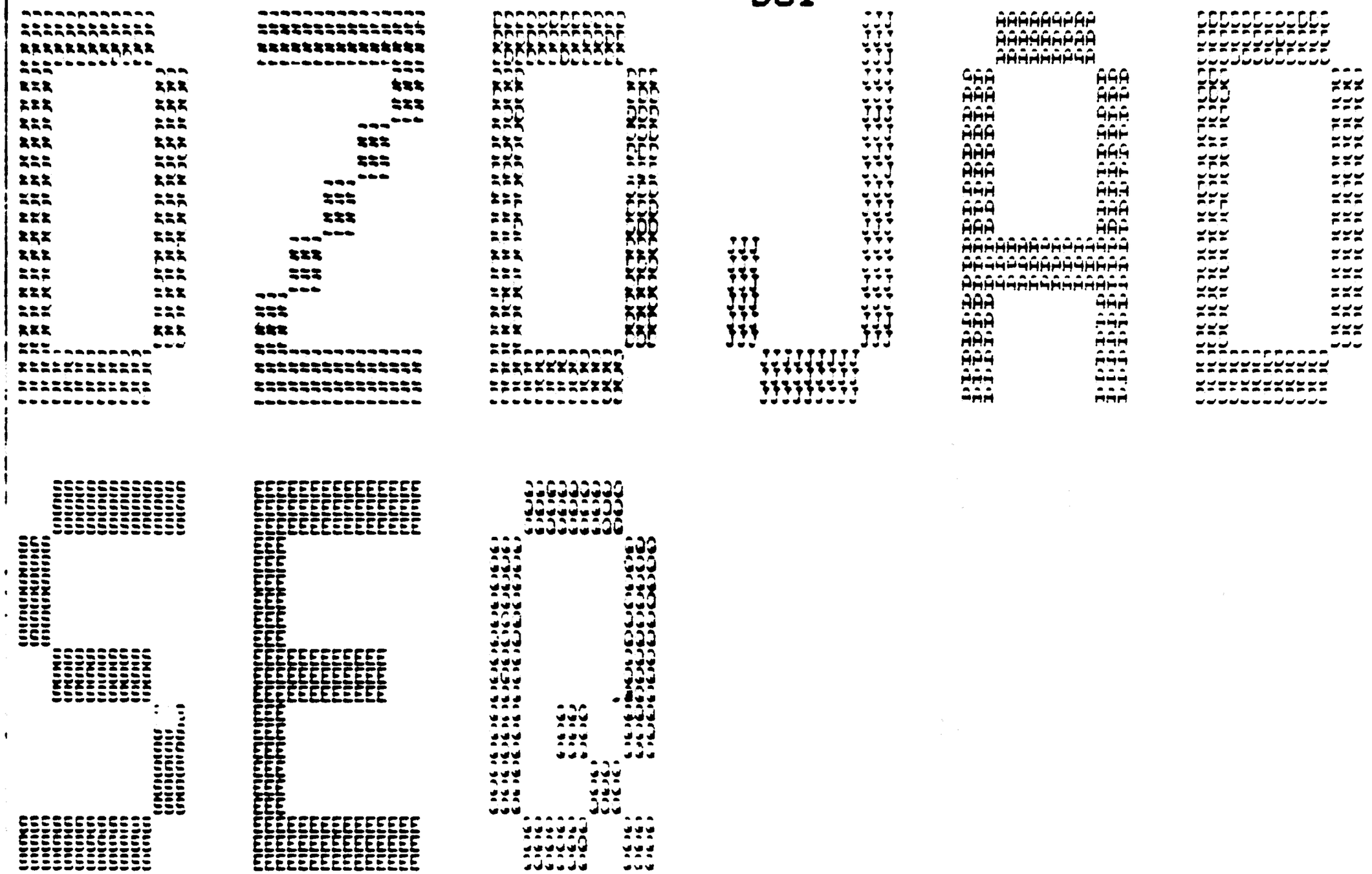







C01

D01




```
_PTSS  Version 6(1003)-4,  Running on MTA1
*SET TIME/DATE 13-Oct-78 15:13:22
Previous Crash at 13-Oct-78  14:51:19
```


Monitor IPC-D SCP3 [1A3] *START*

DECnet Bas. 33 Gate 13-Oct-78 15:13:22

Distribution to ML21-4, slot 134


```
ML21-4:CM.IES -- Distribution to ML21-4, slot 134
```


15:13:27

.REM *

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDJA-D-D
PRODUCT NAME: DJ11 LOGIC TESTS
PROGRAM DATE: MAY 1976
MAINTAINER: DIAGNOSTIC GROUP

TABLE OF CONTENTS

CONTENTS

DESCRIPTION

1. ABSTRACT

THIS PROGRAM TESTS THE LOGIC OF THE DJ11 ASYNCHRONOUS MULTIPLEXER IN MAINTENANCE MODE. IT CHECKS THAT ALL CONTROL REGISTERS FUNCTION PROPERLY, THAT INTERRUPTS OCCUR 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-D2DJB PROGRAMS 2 AND 3 FOR ON-LINE TESTING. THE PROGRAM SHOULD BE RUN FOR AT LEAST 2 PASSES WITH ALL SWITCHES DOWN.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD COMPUTER WITH CONSOLE TELETYPE
UP TO 16 DJ11 ASYNCHRONOUS 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.
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. ALTERNATE PASS WILL RUN WITH THE 7-BIT SET.

THE SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<14> = 1 ..... SCOPE LOOP
SW<13> = 1 ..... INHIBIT PRINTOUT
SW<12> = 1 .....  INHIBIT TRACE TRAPPING
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

192 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(13) ON A 1.

5.2.4 TRTRAP

IF SW(12) IS ON A 0, THE T-BIT WILL BE SET ON ALTERNATE PASSES. WHEN THE T-BIT IS SET, THE PROCESSOR TRAPS AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTI" OR "RTT" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTIONS. THIS SEQUENCE IS CONTINUED UNTIL THE END OF THE PROGRAM IS REACHED.

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 "P(REVIOUS)" SELECTS THE NUMBER OF UNITS PREV SLY SELECTED.

4) "STANDARD CONFIGURATION? "
"Y(ES)" OR DEFAULT (CAR: AGE 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" (LOC 1202). 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 8 DJ11'S AT ONCE, THE EXICUTION 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

```
   363                                  150 00:13:00
   364                                  300 00:05:30
```

[illegible]

[illegible]

8.2 STACK POINTER

STACK IS INITIALLY SET TO 1200

8.3 PASS COUNT

A 32 BIT (2 WORD) PASS COUNT IS KEPT IN "PASSES". THIS [illegible] BE CLEARED FROM THE CONSOLE. BY [illegible] AT 200, OR BY RELOADING THE PROGRAM.

8.4 POWER FAIL

[illegible]

9. PROGRAM DESCRIPTION

[illegible]

```
.TITLE MAINDEC-11-D2DJA-D       DZ11 LOGIC TESTS
.ENABLE ABS
.ENABLE AMA
;COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
;PROGRAM BY KEN CHAPMAN

;     SWITCH                    USE
;     ------                    ---
;       15      100000          ;HALT ON ERROR
;       14       40000          ;LOOP ON TEST
;       13       20000          ;INHIBIT ERROR TYPEOUTS
;       12       10000          ;INHIBIT ERROR TRAP
;       11        4000          ;INHIBIT ITERATIONS
;       10        2000          ;BELL ON PASS COMPLETE
;        9        1000          ;BELL ON ERROR
;        8         400          ;LOOP ON ERROR
;        7         200          ;LOOP ON TEST IN SW 7:0
;        6         100
;        5          40
;        4          20
;        3          10
;        2           4
;        1           2
;        0           1

DZ11 REGISTER BIT ASSIGNMENTS:

CONTROL STATUS REGISTER (CSR) XXXXX0
        BIT15   RECEIVER ENABLE (READ/WRITE)
        BIT14   TRANSMIT DOUBLE BUFFER ENABLE (READ/WRITE)
        BIT13   MAINTENANCE (READ/WRITE)
        BIT12   CLEAR MSS (WRITE ONLY)
        BIT11   CLEAR MSS FLAG (READ ONLY)
        BIT10   NOT USED
        BIT09   RECEIVER INTERRUPT ENABLE (READ/WRITE)
        BIT08   DONE (READ ONLY)
        BIT07   MASTER TRANSMITTER SCAN ENABLE (READ/WRITE)
        BIT06   NOT USED
        BIT05   READ/WRITE BREAK REGISTER (READ/WRITE)
        BIT04   NOT USED
        BIT03   STATUS ENABLE (READ/WRITE)
        BIT02   SILO OVERRUN (READ ONLY)
        BIT01   MASTER TRANSMITTER INTERRUPT ENABLE (READ/WRITE)
        BIT00   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) XXXX04 (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 BSR SET) (READ/WRITE)

BIT0-15 TRNSMIT A BREAK ON CORRESPONDING LINE!

```
                                .       =       0                       ;TRAP CATCHER IN LOCATIONS 0 THRU 776
000000  000000  000000          .BLKW   2.0                             ;LOCATIONS 0 AND 2 CONTAIN "HALT" INSTRUCTIONS
                                ;LOCATIONS 4 THRU 56 CONTAIN ".+2" AND "HALT" IN EVERY VECTOR
                                ;LOCATIONS 60 THRU 776 CONTAIN ".+2" AND "IOT" IN EVERY VECTOR


                                .       =       46
000046  016516                  .WORD   $ENDAD

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

                                .       =       200
000200  000137  001334          JMP     BEGIN                   ;200 ALWAYS IS THE STARTING ADDRESS
                                .       =       +2              ;RESTART ADDRESS
000202  000137  002302          JMP     RESTAR

                                .       =       1200
001200  000000                  ICNT:   0                       ;ITERATION COUNT-LH, TEST NO.-RH
001202  000000                  ERRORS: 0                       ;ERROR COUNT REGISTER
001204  000000  000000          PCNT:   0.0                     ;PASS COUNT REGISTER

001206  160010                  CSR:    160010                  ;CONTROL STATUS REGISTER(DJ UNDER TEST)
001210  160012                  RBUF:   160012                  ;RECEIVER BUFFER REGISTER(DJ UNDER TEST)
                                *CR:                            ;TRANSMITTER CONTROL REGISTER(DJ UNDER TEST)
001212  160014                  BRKSR:  160014                  ;BREAK STATUS REGISTER(DJ UNDER TEST)
001214  160016                  *BUF:   160016                  ;TRANSMITTER BUFFER REGISTER(DJ UNDER TEST)

001216  000300                  RCVVEC: 300                     ;RECEIVER INTERUPT VECTOR ADDRESS(DJ UNDER TEST
001220  000302                  RCVLVL: 302
001222  000304                  XMTVEC: 304                     ;TRANSMITTER INTERUPT VECTOR ADDRESS(DJ UNDER TEST
001224  000306                  XMTLVL: 306

001226  160010                  DEVADR: 160010                  ;FIRST DEVICE ADDRESS(SELECTED)
001230  000300                  VECADR: 300                     ;FIRST VECTOR ADDRESS(SELECTED)
001232  000001                  UNITS:  1                       ;NUMBER OF UNITS TO BE TESTED
001234  000040  000000          LENGTH: .BLKB   40              ;TABLE OF CHARACTER LENGTHS (MASK)
001276  000004  000000          PARITY: .BLKB   4               ;TABLE OF ODD PARITY FLAGS
001304  000000                  TIMER:  0                       ;USED TO SAVE RELATIVE TIMES

001306  000000                  DJUNIT: 0                       ;UNIT NUMBER OF DJ11 UNDER TEST
001310  000000                  CHLEN:  0                       ;CHAR MASK TABLE POINTER
001312  000000                  CHPAR:  0                       ;UNIT FLAG (FOR ODD PARITY FLAG)
001314  160010                  DEFADR: 160010                  ;DEFAULT FIRST DEVICE ADDRESS
001316  000300                  DEFVEC: 300                     ;DEFAULT FIRST VECTOR ADDRESS

001320  177570                  SWR:    177570
001322  177570                  DISP_AT:177570

001324  177777                  FTIME:  -1
```

```
001200                  BEGIN:  MOV     #ICNT,  SP      ;SET UP STACK POINTER
001206 004537 016216            JSR     PC,     SUSWRR  ;CHECK FOR SWITCH REGISTER
001212 012700 000014            MOV     #14,R0
001216 012720 015534            MOV     #YESRT, (R0)+   ;TRACE TRAP VECTOR (14)
001222 012720 000340            MOV     #340,   (R0)+
001226 012720 017104            MOV     #IOTRAP,(R0)+   ;IOT VECTOR (20)
001232 012720 000340            MOV     #340,   (R0)+
001236 012720 016446            MOV     #PDOWN$,(R0)+   ;POWER FAIL VECTOR (24)
001242 012720 000340            MOV     #340,   (R0)+
001246 012720 015774            MOV     #EMT$,  (R0)+   ;EMT VECTOR (30)
001252 012720 000340            MOV     #340,   (R0)+
001256 012720 015632            MOV     #TRAP$, (R0)+   ;TRAP VECTOR (34)
001262 012720 000340            MOV     #340,   (R0)+
001266 012737 001446 000010     MOV     #1$,@#10
001274 006700                   SXT     R0              ;CHECK FOR PDP-11/40 OR 45
001276 012737 000006 015534     MOV     #RTT,@#YESRT
001304 012737 000006 000006     MOV     #RTT,@#6
001312 012737 000400 177774     MOV     #400,@#177774   ;SET UP STACK LIMIT TO 1000
001320 005037 000006            CLR     @#6
001324 012737 000012 000010 1$: MOV     #12,@#10
001332 005037 001202            CLR     ERRORS          ;CLEAR ERROR COUNTER
001336 005037 001204            CLR     PCNT            ;CLEAR PASS COUNTER
001342 005037 001206            CLR     PCNT+2
001346 005737 000042            TST     @#42            ;CHECK FOR ACT11 OR DDP PRESENT
001352 001404                   BEQ     GETADR          ;BRANCH IF NONE
001354 004737 017470            JSR     PC,     AUTO    ;GO TO SUBROUTINE TO "MAP" DJ11 ON THE SYS
001360 000137 002302            JMP     RESTAR          ;SKIP OPERATOR ACTION

001364 000004 017244    GETADR: TYPE,   MSGADR          ;TYPE "FIRST DJ11 ADDRESS"
001370 004537 016216            JSR     R5,     READIN  ;READ INPUT FROM TTY AND SAVE
001374 001230                   .WORD   DEVADR          ; IN DEVADR
001376 001372                   BNE     GETADR          ;BRANCH IF BAD INPUT
001400 005737 001230            TST     DEVADR
001404 001003                   BNE     1$
001406 013737 001312 001230     MOV     DEFADR,DEVADR
001414 042737 000007 001230 1$: BIC     #7,DEVADR
001422 022737 160000 001230     CMP     #160000,DEVADR
001430 101355                   BHI     GETADR

001432 000004 017274    GETVEC: TYPE,   MSGVEC          ;TYPE "VECTOR ADDRESS:"
001436 004537 016216            JSR     R5,     READIN  ;READ INPUT FROM TTY AND SAVE
001442 001232                   .WORD   VECADR          ;IN VECADR
001444 001372                   BNE     GETVEC          ;BRANCH IF BAD INPUT
001446 005737 001232            TST     VECADR          ;CHECK FOR CR
001452 001003                   BNE     1$
001454 012737 000300 001232     MOV     #300,   VECADR  ;SET TO FIRST FLOATING VECTOR
001462 042737 000007 001232 1$: BIC     #7,     VECADR  ;CLEAR TO MODULO 10
001470 022737 000300 001232     CMP     #300,   VECADR  ;CHECK FOR LOW LIMIT
001476 003355                   BGT     GETVEC
001500 022737 001000 001232     CMP     #1000,  VECADR  ;CHECK FOR UPPER LIMIT
001506 002375                   BLE     GETVEC
001510 000004 017320    GETNUM: TYPE,   MSGNUM  ;TYPE"NUMBER OR UNITS:"
001514 004737 016364            JSR     PC,READS$       ;READ INPUT FROM THE TTY
                        ; CHECK STRING FOR VALID DIGITS,TERMINATOR,AND 'P'
001520 012702 016512            MOV     #INPUT,R2       ;POINTS TO INPUT STRING
```

```
001646  112201                          MOVB    (R2)+,R1        ;LOAD 1ST CHAR
001650  001434                          BEQ     1$              ;BRANCH IF IMMEDIATE TERMINATOR
001652  122701  000120                  CMPB    #'P,R1          ;CHECK FOR A P
001656  001443                          BEQ     CONFIG          ;CONFIGURE MODE IF SO
001660  120127  000071                  CMPB    R1,#71          ;MUST BE A VALID ASCII NUMBER
001664  101362                          BHI     GETNUM          ;ELSE RETRY
001666  162701  000060                  SUB     #60,R1          ;STRIP ASCII CODE
001672  003754                          BLE     GETNUM          ;SHOULDN'T BE ZERO OR NEG
                                ;1ST CHAR IS A VALID DIGIT, 1 THRU 9 - CHECK REST OF STRING
001674  105722                          TSTB    (R2)+           ;2ND CHAR A TERMINATOR?
001676  001403                          BEQ     3$              ;YES - R1 HAS THE FINAL # OF UNITS
001700  105712                          TSTB    (R2)            ;NO - NXT CHAR MUST BE THE TERMINATOR
001702  001360                          BNE     GETNUM          ;ELSE RETRY.
001704  120227  000071                  CMPB    -(R2),#71       ;CHECK 2ND CHAR FOR A VALID NUMBER
001710  101360                          BHI     GETNUM
001712  116203                          MOVB    (R2),R3         ;MAKE IT MORE ACCESSIBLE
001714  162703  000060                  SUB     #60,R3          ;STRIP ASCII CODE
001720  100744                          BMI     GETNUM          ;SHOULDN'T BE NEGATIVE
                                ;BOTH LOW AND HIGH ORDER DIGITS ARE OK - CONVERT DECIMAL VALUE.
001722  010146                          MOV     R1,-(SP)        ;SAVE IT FOR LATER
001724  006301                          ASL     R1              ;MULT HI ORDER BY 8
001726  006301                          ASL     R1
001730  006301                          ASL     R1
001732  006316                          ASL     (SP)            ;MULTIPLY IT BY 2
001734  062601                          ADD     (SP)+,R1        ;RESULT IS (HI ORD)*10.
001736  060301                          ADD     R3,R1           ;NOW ADD IN THE LOW ORDER
001740  000402                          BR      3$
001742  012701  000001          1$:     MOV     #1,R1           ;LOAD DEFAULT VALUE HERE
                                ;R1 HAS THE CONVERTED VALUE - COMPARE AGAINST MAX ALLOWED.
001746  020127  000020          3$:     CMP     R1,#DJMXNO      ;TOO BIG?
001752  101327                          BHI     GETNUM          ;YES - RETRY.
001754  010137  001234                  MOV     R1,UNITS        ;VALUE OK - STORE RESULT
001760  012737  000100  001304          MOV     #100.,DJUUT     ;"PRIME" UNIT UNDER TEST NUMBER

001766  001304  017343          CONFIG: TYPE.   MSGCON          ;TYPE "STANDARD CONFIG?"
001772  004737  016364                  JSR     PC,     READ$   ;READ INPUT FROM TTY
001776  122737  000120  016512          CMPB    #'P,    INPUT   ;CHECK FOR "P"
002004  001536                          BEQ     RESTAR          ;BRANCH IF PREVIOUS
002006  012700  000022                  MOV     #22.,   R0      ;SET UP COUNTER
002012  012701  001236                  MOV     #LENGTH,R1      ;POINT TO CHAR LEN TABLE
002016  005021                  1$:     CLR     (R1)+           ;PUT CHAR MASK FOR 9 IN CHAR TABLE
                                                                ;AND CLR PARITY TABLE
002020  005300                          DEC     R0              ;COUNT DOWN
002022  001375                          BNE     1$              ;BRANCH IF NOT DONE
002024  105737  016512                  TSTB    INPUT           ;CHECK FOR CR
002030  001524                          BEQ     RESTAR          ;BRANCH IF DEFAULT
002032  122737  000131  016512          CMPB    #131,   INPUT   ;CHECK FOR "Y"
002040  001520                          BEQ     RESTAR          ;BRANCH IF DEFAULT
002042  122737  000116  016512          CMPB    #116,   INPUT   ;CHECK FOR "N"
002050  001346                          BNE     CONFIG          ;BRANCH IF ILLEGAL ENTRY
002052  005000                          CLR     R0              ;CLR UNIT COUNTER
002054  005001                          CLR     R1              ;CLR LINE COUNTER
002056  012702  001236                  MOV     #LENGTH,R2      ;SET UP POINTER TO CHAR MASK TABLE
002062  012703  001276                  MOV     #PARITY,R3      ;SET UP POINTER TO PARITY TABLE
002066  012704  000001                  MOV     #1,R4           ;SET UP MARKER
002072  005004  017377          TYPLIN: TYPE.   MSGLIN          ;TYPE "LINES "
```

```
002376  010105                      MOV     R1,TTY          ;TYPE R1 IN OCTAL
002400  004737  016572               JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
002404  105004  016410               .TYPE,  MSGDAS          ;TYPE A DASH (-)
002410  062701  000003               ADD     #3.     R1      ;LAST LINE IN GROUP OF 4
002414  010105                      MOV     R1,TTY          ;TYPE R1 IN OCTAL
002416  004737  016572               JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
002422  005201                      INC     R1              ;FIRST LINE NEXT GROUP
002424  105004  017414       GETLEN: .TYPE,  MSGLEN          ;TYPE "CHAR LENGTH: "
002430  004737  016364               JSR     PC,     READ$   ;READ FROM THE TTY
002434  105737  016512               TSTB    INPUT           ;CHECK FOR CR (DEFAULT)
002440  001421                      BEQ     GETPAR          ;BRANCH IF DEFAULT
002442  105737  016513               TSTB    INPUT+1         ;CHECK FOR BAD DATA
002446  001366                      BNE     GETLEN          ;BRANCH IF BAD
002450  162737  000060  016513       SUB     #60.    INPUT   ;CONVERT FROM ASCII
002456  112712  177777               MOVB    #-1,    (R2     ;SET UP MASK
002462  106312               2$:    ASLB    (R2)            ;CLEAR MASK BIT BY BIT
002464  103357                      BCC     GETLEN          ;BAD INPUT, > 8
002466  105337  016512               DECB    INPUT           ;COUNT
002472  001373                      BNE     2$              ;BRANCH IF NOT DONE
002474  132712  000037               BITB    #37,    (R2)    ;CHECK FOR 5 CHAR
002500  001351                      BNE     GETLEN          ;BAD INPUT, < 5
002502  005202                      INC     R2              ;INC TO NEXT LINE GROUP
002504  105004  017435       GETPAR: .TYPE,  MSGPAR          ;TYPE "PARITY (NO, ODD, EVEN):
002510  004737  016364               JSR     PC,READ$        ;READ INPUT FROM TTY
002514  005737  016512               TST     INPUT           ;CHECK FOR CR
002520  001415                      BEQ     3$              ;BRANCH IF DEFAULT
002522  122737  000116  016512       CMPB    #116.INPUT      ;CHECK FOR "N"
002530  001411                      BEQ     3$              ;BRANCH IF "NO"
002532  122737  000105  016512       CMPB    #105.INPUT      ;CHECK FOR "E"
002540  001405                      BEQ     3$              ;BRANCH IF "EVEN"
002542  122737  000117  016512       CMPB    #117.INPUT      ;CHECK FOR "O"
002550  001355                      BNE     GETPAR          ;BRANCH IF NONE
002552  150413                      BISB    R4,     (R3)    ;SET THE ODD PARITY FLAG
002554  005203               3$:    INC     R3              ;UP DATE PARITY TABLE POINTER
002556  032701  000017               BIT     #17,R1          ;CHECK FOR NEXT UNIT
002562  001303                      BNE     TYPLIN          ;BRANCH IF NOT
002564  012703  001276               MOV     #PARITY,R3      ;RESET PARITY TABLE POINTER
002570  006304                      ASL     R4              ;FLAG TO NEXT UNIT
002572  005200                      INC     R0              ;COUNT UNITS
002574  020037  001234               CMP     R0,UNITS        ;CHECK FOR LAST
002600  001274                      BNE     TYPLIN          ;BRANCH IF MORE
```

```
                                                
002302  000005                  RESTAR: RESET                           ;ISSUE RESET
002304  012706  001230                  MOV     #ICNT,  SP              ;SET UP STACK POINTER
002310  005737  001322                  TST     FTIME
002314  001410                          BEQ     CLRVEC
002316  022737  000176  001316          CMP     #SWREG, SWR
002324  001004                          BNE     CLRVEC
002326  004737  017770                  JSR     PC,     CNTLU
002332  005037  001322                  CLR     FTIME
002336  012700  000300          CLRVEC: MOV     #300,R0                 ;BEGINNING OF FLOATING VECTORS
002342  005720                  1$:     TST     (R0)+
002344  010060  177776                  MOV     R0,-2(R0)               ;".+2"
002350  012720  000004                  MOV     #IOT,   (R0)+           ;"IOT"
002354  022700  001000                  CMP     #1000,R0
002360  001370                          BNE     1$
002362  062737  000010  001210          ADD     #10,    CSR             ;UPDATE DEVICE ADDRESS TO NEXT UNIT
002370  062737  000010  001220          ADD     #10,    RCVVEC          ;UPDATE DEVICE VECTOR ADDRESS
002376  062737  000004  001306          ADD     #4,     DJLEN           ;MOVE CHAR TABLE POINTER
002404  006337  001310                  ASL     DJPAR                   ;UPDATE PARITY FLAG MARKER
002410  023737  001304  001234          CMP     DJUUT,UNITS
002416  002416                          BLT     2$
002420  005037  001304                  CLR     DJUUT
002424  013737  001230  001210          MOV     DEVADR,CSR
002432  013737  001232  001220          MOV     VECADR,RCVVEC
002440  012737  001236  001306          MOV     #LENGTH,DJLEN           ;INIT CHAR TABLE POINTER
002446  012737  000001  001310          MOV     #1,DJPAR                ;INIT PARITY FLAG MARKER
002454  005237  001304          2$:     INC     DJUUT
002460  013701  001210                  MOV     CSR,    R1              ;SET UP ALL THE REGISTER ADDRESSES
002464  062701  000002                  ADD     #2,     R1              ;ADD 2
002470  010137  001212                  MOV     R1,     RBUF            ;SET UP RECEIVER BUFFER
002474  062701  000002                  ADD     #2,     R1              ;ADD 2
002500  010137  001214                  MOV     R1,     TCR             ;SET UP TRANSMITTER CONTROL REG
                                                                        ; AND BREAK STATUS REG
002504  062701  000002                  ADD     #2,     R1              ;ADD 2
002510  010137  001216                  MOV     R1,     TBUF            ;SET UP TRANSMITTER BUFFER
002514  013701  001220                  MOV     RCVVEC, R1              ;POINTER FOR VECTOR SETUP
002520  005721                          TST     (R1)+                   ;INC R1
002522  010137  001222                  MOV     R1,     RCVLVL          ;SET INT LVL
002526  005721                          TST     (R1)+                   ;INC R1
002530  010137  001224                  MOV     R1,     XMTVEC          ;TRANSMITTER VECTOR
002534  005721                          TST     (R1)+                   ;INC R1
002536  010137  001226                  MOV     R1,     XMTLVL          ;XMT INT LVL ADR
002542  005037  001200                  CLR     ICNT
002546  005037  015770                  CLR     LAD
002552  104400                          SCOPE
```

```
                                ;*****************************************************************
                                ; TEST 1:       TEST ABILITY TO REFERENCE CSR WITHOUT TRAPPING
                                ;               AND ABILITY TO CLR CSR.
                                ;PROBABLE FAULTY LOGIC:  M105; M7285 (D2-6) E29
                                ;*****************************************************************

002554  012737  002612  000004  TST1:  MOV     #1$,    @#4     ;SET UP TIME-OUT TRAP VECTOR
002562  012737  000340  000006         MOV     #LEVEL7,@#6
002570  005777  176414                 TST     @CSR            ;REFERENCE CSR  (READ)
002574  005077  176410                 CLR     @CSR            ;CLEAR CSR      (WRITE)
002600  005577  176404                 ADC     @CSR            ;CHECK CSR      (READ AND WRITE)
002604  001405                         BEQ     2$              ;BRANCH IF OK
002606  104000                         HLT                     ;CSR NOT CLEARED

002610  000403                         BR      2$              ;SKIP ISR

002612  012601                 1$:     MOV     (SP)+,R1        ;SAVE RTI ADR FOR TYPING
002614  104001                         HLT+1                   ;CAN'T REFERENCE CSR!

002616  005726                         TST     (SP)+           ;FINISH CLEARING STACK
002620  012737  000006  000004  2$:     MOV     #6,@#4
002626  005037  000006                 CLR     @#6
002632  104400                         SCOPE

                                ;*****************************************************************
                                ;TEST 2:        TEST THAT CSR BIT1 CAN BE SET AND CLEARED
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E25,E7, (D2-4) E47,E64
                                ;*****************************************************************

002634  012777  000002  176346  TST2:  MOV     #BIT1,@CSR      ;SET BIT1
002642  032777  000002  176340         BIT     #BIT1,@CSR      ;CHECK THAT BIT1 IS SET
002650  001001                         BNE     .+4             ;BRANCH IF OK
002652  104000                         HLT                     ;CSR BIT1 FAILED TO SET

002654  032777  177775  176326         BIT     #177775,@CSR    ;CHECK THAT NO OTHER BIT SET
002662  001401                         BEQ     .+4             ;BRANCH IF Ok
002664  104000                         HLT                     ;EXTRA BIT SET IN CSR

002666  005077  176316                 CLR     @CSR            ;CLEAR BIT1
002672  032777  000002  176310         BIT     #BIT1,@CSR      ;CHECK THAT BIT1 IS CLEARED
002700  001401                         BEQ     .+4             ;BRANCH IF OK
002702  104000                         HLT                     ;CSR BIT1 FAILED TO CLEAR

002704  104400                         SCOPE

                                ;*****************************************************************
                                ;TEST 3:        TEST THAT CSR BIT2 CAN BE SET AND CLEARED
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E25,E7, (D2-4) E47,E64
                                ;*****************************************************************

002706  012777  000004  176274  TST3:  MOV     #BIT2,@CSR      ;SET BIT2
002714  032777  000004  176266         BIT     #BIT2,@CSR      ;CHECK THAT BIT2 IS SET
002722  001001                         BNE     .+4             ;BRANCH IF OK
002724  104000                         HLT                     ;CSR BIT2 FAILED TO SET
```

```
002726  032777  177773  176254          BIT     #177773,@CSR    ;CHECK THAT NO OTHER BIT SET
002734  001401                          BEQ     .+4             ;BRANCH IF OK
002736  104000                          HLT                     ;EXTRA BIT SET IN CSR

002740  005077  176244                  CLR     @CSR            ;CLEAR BIT2
002744  032777  000004  176236          BIT     #BIT2,@CSR      ;CHECK THAT BIT2 IS CLEARED
002752  001401                          BEQ     .+4             ;BRANCH IF OK
002754  104000                          HLT                     ;CSR BIT2 FAILED TO CLEAR

002756  104400                          SCOPE

                        ;*****************************************************************
                        ;TEST 4:        TEST THAT CSR BIT6 CAN BE SET AND CLEARED
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E4,E18, (D2-4) E47,E54
                        ;*****************************************************************

002760  012777  000100  176222  TST4:   MOV     #BIT6,@CSR      ;SET BIT6
002766  032777  000100  176214          BIT     #BIT6,@CSR      ;CHECK THAT BIT6 IS SET
002774  001001                          BNE     .+4             ;BRANCH IF OK
002776  104000                          HLT                     ;CSR BIT6 FAILED TO SET

003000  032777  177677  176202          BIT     #177677,@CSR    ;CHECK THAT NO OTHER BIT SET
003006  001401                          BEQ     .+4             ;BRANCH IF OK
003010  104000                          HLT                     ;EXTRA BIT SET IN CSR

003012  005077  176172                  CLR     @CSR            ;CLEAR BIT6
003016  032777  000100  176164          BIT     #BIT6,@CSR      ;CHECK THAT BIT6 IS CLEARED
003024  001401                          BEQ     .+4             ;BRANCH IF OK
003026  104000                          HLT                     ;CSR BIT6 FAILED TO CLEAR

003030  104400                          SCOPE

                        ;*****************************************************************
                        ;TEST 5:        TEST THAT CSR BIT8 CAN BE SET AND CLEARED
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E6,E18, (D2-4) E47,E31
                        ;*****************************************************************

003032  012777  000400  176150  TST5:   MOV     #BIT8,@CSR      ;SET BIT8
003040  032777  000400  176142          BIT     #BIT8,@CSR      ;CHECK THAT BIT8 IS SET
003046  001001                          BNE     .+4             ;BRANCH IF OK
003050  104000                          HLT                     ;CSR BIT8 FAILED TO SET

003052  032777  177377  176130          BIT     #177377,@CSR    ;CHECK THAT NO OTHER BIT SET
003060  001401                          BEQ     .+4             ;BRANCH IF OK
003062  104000                          HLT                     ;EXTRA BIT SET IN CSR

003064  005077  176120                  CLR     @CSR            ;CLEAR BIT8
003070  032777  000400  176112          BIT     #BIT8,@CSR      ;CHECK THAT BIT8 IS CLEARED
003076  001401                          BEQ     .+4             ;BRANCH IF OK
003100  104000                          HLT                     ;CSR BIT8 FAILED TO CLEAR

003102  104400                          SCOPE

                        ;*****************************************************************
                        ;TEST 6:        TEST THAT CSR BIT10 CAN BE SET AND CLEARED
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E30,E24, (D2-4) E47,E31
```

```
                                ;*****************************************************************
                                ;
003104  012777  002000  176076  TST6:   MOV     #BIT10,@CSR     ;SET BIT10
003112  032777  002000  176070          BIT     #BIT10,@CSR     ;CHECK THAT BIT10 IS SET
003120  001001                          BNE     .+4             ;BRANCH IF OK
003122  104000                          HLT                     ;CSR BIT10 FAILED TO SET
                                ;
003124  032777  175777  176056          BIT     #175777,@CSR    ;CHECK THAT NO OTHER BIT SET
003132  001401                          BEQ     .+4             ;BRANCH IF OK
003134  104000                          HLT                     ;EXTRA BIT SET IN CSR
                                ;
003136  005077  176046                  CLR     @CSR            ;CLEAR BIT10
003142  032777  002000  176040          BIT     #BIT10,@CSR     ;CHECK THAT BIT10 IS CLEARED
003150  001401                          BEQ     .+4             ;BRANCH IF OK
003152  104000                          HLT                     ;CSR BIT10 FAILED TO CLEAR
                                ;
003154  104400                          SCOPE
                                ;
                                ;*****************************************************************
                                ;TEST 7:        TEST THAT CSR BIT12 CAN BE SET AND CLEARED
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5,E1, (D2-4) E47,E31
                                ;*****************************************************************
                                ;
003156  012777  010000  176024  TST7:   MOV     #BIT12,@CSR     ;SET BIT12
003164  032777  010000  176016          BIT     #BIT12,@CSR     ;CHECK THAT BIT12 IS SET
003172  001001                          BNE     .+4             ;BRANCH IF OK
003174  104000                          HLT                     ;CSR BIT12 FAILED TO SET
                                ;
003176  032777  167777  176004          BIT     #167777,@CSR    ;CHECK THAT NO OTHER BIT SET
003204  001401                          BEQ     .+4             ;BRANCH IF OK
003206  104000                          HLT                     ;EXTRA BIT SET IN CSR
                                ;
003210  005077  175774                  CLR     @CSR            ;CLEAR BIT12
003214  032777  010000  175766          BIT     #BIT12,@CSR     ;CHECK THAT BIT12 IS CLEARED
003222  001401                          BEQ     .+4             ;BRANCH IF OK
003224  104000                          HLT                     ;CSR BIT12 FAILED TO CLEAR
                                ;
003226  104400                          SCOPE
                                ;
                                ;*****************************************************************
                                ;TEST 10:       TEST THAT CSR BIT14 CAN BE SET AND CLEARED
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E3,E1, (D2-4) E47,E31
                                ;*****************************************************************
                                ;
003230  012777  040000  175752  TST10:  MOV     #BIT14,@CSR     ;SET BIT14
003236  032777  040000  175744          BIT     #BIT14,@CSR     ;CHECK THAT BIT14 IS SET
003244  001001                          BNE     .+4             ;BRANCH IF OK
003246  104000                          HLT                     ;CSR BIT14 FAILED TO SET
                                ;
003250  032777  137777  175732          BIT     #137777,@CSR    ;CHECK THAT NO OTHER BIT SET
003256  001401                          BEQ     .+4             ;BRANCH IF OK
003260  104000                          HLT                     ;EXTRA BIT SET IN CSR
                                ;
003262  005077  175722                  CLR     @CSR            ;CLEAR BIT14
003266  032777  040000  175714          BIT     #BIT14,@CSR     ;CHECK THAT BIT14 IS CLEARED
003274  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
003276  104000                          HLT                     ;CSR BIT14 FAILED TO CLEAR

003300  104400                          SCOPE

                        ;*****************************************************************
                        ;TEST 11:       TEST THAT RECEIVER ENABLE (BIT0 OF THE CSR, CAN BE SET
                        ;                 AND CLEARED, AND THAT CLEAR MOS (BIT3) IS WRITE ONLY.
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E26,E36,E7,E24, (D2-8) E17,E14,E15
                        ;*****************************************************************

003302  012777  000004  175700  TST11:  MOV     #BIT2,  @CSR    ;SET MAINTENANCE MODE (BIT2)
003310  005005                          CLR     R5              ;SET UP COUNTER
003312  052777  000010  175670          BIS     #BIT3,  @CSR    ;SET CLEAR MOS (BIT3)
003320  017701  175664          1$:     MOV     @CSR,   R1      ;SAVE CSR
003324  032701  000010                  BIT     #BIT3,  R1      ;CHECK CLEAR MOS (BIT3)
003330  001401                          BEQ     .+4             ;BRANCH IF OK
003332  104001                          HLT+1                   ;CLEAR MOS (BIT3) SET (WRITE-ONLY)
                                                                ;R1 = CONTENTS OF CSR
003334  032701  000020                  BIT     #BIT4,  R1      ;CHECK CLEAR MOS FLAG
003340  001403                          BEQ     2$              ;BRANCH IF CLEARED
003342  105305                          DECB    R5              ;WAIT FOR MOS TO CLEAR
003344  001365                          BNE     1$              ;BRANCH IF MORE TIME
003346  104001                          HLT+1                   ;CLEAR MOS FLAG (BIT4) FAILED TO CLEAR
                                                                ;R1 = CONTENTS OF CSR
003350  022701  000004          2$:     CMP     #BIT2,  R1      ;CHECK THAT ONLY MAINTENANCE BIT SET
003354  001401                          BEQ     .+4             ;BRANCH IF OK
003356  104001                          HLT+1                   ;CLEAR MOS CLEARED MAINTENANCE
                                                                ; OR SET OTHER CSR BITS
                                                                ;R1 = CONTENTS OF CSR
003360  052777  000001  175622          BIS     #BIT0,  @CSR    ;SET RECEIVER ENABLE
003366  017701  175616                  MOV     @CSR,   R1      ;SAVE CSR
003372  032777  000001  175610          BIT     #BIT0,  @CSR    ;CHECK THAT RECEIVER ENABLE SET
003400  001001                          BNE     .+4             ;BRANCH IF OK
003402  104001                          HLT+1                   ;RECEIVER ENABLE FAILED TO SET
                                                                ;R1 = CONTENTS OF CSR
003404  022777  000005  175576          CMP     #5,     @CSR    ;CHECK REST OF CSR
003412  001401                          BEQ     .+4             ;BRANCH IF OK
003414  104001                          HLT+1                   ;CSR ERROR
                                                                ;R1 = CONTENTS OF CSR
                        ;NOTE:  IF THE TTY MODULE IS BEING USED AND DONE (BIT7) IS SET,
                        ;       THE ERROR COULD BE DUE TO MAINTENANCE OR CLEAR MOS NOT WORKING.
003416  042777  000001  175564          BIC     #BIT0,  @CSR    ;CLEAR RECEIVER ENABLE
003424  017701  175560                  MOV     @CSR,   R1      ;SAVE CSR
003430  022777  000004  175552          CMP     #BIT2,  @CSR    ;CHECK CSR
003436  001401                          BEQ     .+4             ;BRANCH IF OK
003440  104001                          HLT+1                   ;RECEIVER ENABLE DIDN'T CLEAR
                                                                ; OR OTHER CSR BIT SET
                                                                ;R1 = CONTENTS OF CSR

003442  104400                          SCOPE

                        ;*****************************************************************
                        ; TEST 12:       TEST THAT CSR RESPONDS PROPERLY TO BYTE COMMANDS
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-4) E47
                        ;*****************************************************************
```

```
003444  012777  052506  175536  TST12:  MOV     #052506,@CSR    ;SET TEST NUMBER IN CSR
003452  105077  175532          CLRB    @CSR            ;CLR EVEN BYTE
003456  017701  175526          MOV     @CSR,   R1      ;SAVE CSR
003462  022701  052400          CMP     #052400,R1      ;CHECK CSR
003466  001401                  BEQ     .+4             ;BRANCH IF OK
003470  104001                  HLT+1                   ;EVEN BYTE CLR FAILED ON CSR
                                                        ;R1 = CONTENTS OF CSR
003472  012777  052506  175510          MOV     #052506,@CSR    ;SET TEST NUMBER IN CSR
003500  005237  001210          INC     CSR             ;INC TO ODD BYTE
003504  105077  175500          CLRB    @CSR            ;CLR ODD BYTE
003510  005337  001210          DEC     CSR             ;RESTORE TO EVEN
003514  017701  175470          MOV     @CSR,   R1      ;SAVE CSR
003520  022701  000106          CMP     #000106,R1      ;CHECK CSR
003524  001401                  BEQ     .+4             ;BRANCH IF OK
003526  104001                  HLT+1                   ;ODD BYTE CLR FAILED ON CSR
                                                        ;R1 = CONTENTS OF CSR

003530  104400                  SCOPE

                        ;*****************************************************************
                        ;TEST 13:       TEST THAT THE BIS AND BIC INSTRUCTIONS SET AND CLEAR R/W
                        ;               BITS OF CSR
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-4) E47
                        ;*****************************************************************

003532  005077  175452  TST13:  CLR     @CSR            ;CLEAR THE CSR
003536  017701  175446          MOV     @CSR,   R1      ;CHECK AND SAVE CSR
003542  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
003544  104001                  HLT+1                   ;RESET FAILED TO CLR CSR

003546  052777  052506  175434          BIS     #052506,        @CSR    ;SET ALL R/W BITS OF CSR
003554  022777  052506  175426          CMP     #052506,        @CSR    ;CHECK THAT THEY GOT SET
003562  001401                  BEQ     .+4             ;BRANCH IF OK
003564  104000                  HLT                     ;REG FAILED CMP

003566  042777  052506  175414          BIC     #052506,        @CSR    ;CLEAR CSR
003574  017701  175410          MOV     @CSR,   R1      ;CHECK AND SAVE CSR
003600  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
003602  104001                  HLT+1                   ;CLR FAILED TO CLR CSR

003604  104400                  SCOPE

                        ;*****************************************************************
                        ;TEST 14:       TEST BITS OF TCR FOR READ/WRITE CAPABILITY
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) ALL, (D2-3) E8,E20,E31,E43,E41
                        ;*****************************************************************

003606  012777  177777  175400  TST14:  MOV     #177777,@TCR    ;SET ALL BITS OF TCR
003614  017701  175374          MOV     @TCR,   R1      ;CHECK AND SAVE TCR
003620  022701  177777          CMP     #177777,R1      ;CHECK THAT ALL THE BITS ARE SET
003624  001401                  BEQ     .+4             ;BRANCH IF OK
003626  104001                  HLT+1                   ;BIT(S) OF TCR FAILED TO SET

003630  005077  175360          CLR     @TCR            ;CLEAR TCR
003634  017701  175354          MOV     @TCR,   R1      ;CHECK THAT IT CLEARED AND SAVE
003640  001401                  BEQ     .+4             ;BRANCH IF CLR
```

```
                                ;BITS OF TCR FAILED TO CLEAR

                        SCOPE

                ;*****************************************************************
                ;TEST 15:       TEST THAT TCR RESPONDS PROPERLY TO BYTE COMMANDS
                ;PROBABLE FAULTY LOGIC:  M7295  D2-3  E41
                ;*****************************************************************

                TST15:  MOV     #177777,@TCR    ;SET TEST NUMBER IN TCR
                        INCB    @TCR            ;CLR EVEN BYTE
                        MOV     @TCR,R1         ;SAVE TCR
                        BIC     #^C400,R1       ;CHECK TCR
                        BEQ     .+4             ;BRANCH IF OK
                        MTE     +1              ;EVEN BYTE CLR FAILED ON TCR
                                                ;R1 = CONTENTS OF TCR
                        MOV     #177777,@TCR    ;SET TEST NUMBER IN TCR
                        INC     @TCR            ;CLR ODD BYTE
                        MOVB    @TCR,R1         ;CLR ODD BYTE
                        BIC     #^C377,R1       ;SAVE TCR
                        BEQ     .+4             ;CHECK TCR
                        MTE     +1              ;BRANCH IF OK
                                                ;ODD BYTE CLR FAILED ON TCR
                                                ;R1 = CONTENTS OF TCR

                        SCOPE

                ;*****************************************************************
                ;TEST 16:       TEST THAT THE BIS AND BIC INSTRUCTIONS SET AND CLEAR A
                ;               BITS OF TCR
                ;PROBABLE FAULTY LOGIC:  M7295  D2-3  E41
                ;*****************************************************************

                ST16:   CLR     @TCR            ;CLEAR THE TCR
                        MOV     @TCR,   R1      ;CHECK AND SAVE TCR
                        BEQ     .+4             ;BRANCH IF CLEARED OK
                        MTE     +1              ;RESET FAILED TO CLR TCR

                        BIS     #[illegible],@TCR   ;SET ALL R-BITS OF TCR
                        BIC     #[illegible],@TCR   ;CHECK THAT THEY GOT SET
                        BEQ     .+4             ;BRANCH IF OK
                        MTE     +1              ;BIS FAILED CMP

                        BIS     #[illegible],@TCR   ;CLEAR TCR
                        MOV     @TCR,   R1      ;CHECK AND SAVE TCR
                        BEQ     .+4             ;BRANCH IF CLEARED OK
                        MTE     +1              ;CLR FAILED TO CLR TCR

                        SCOPE
                        MOV     #1,     TIMER   ;INITIALIZE TIMER
                        MOV     #.+6,   LAD     ;RESET LOOP ADDRESS

                ;*****************************************************************
                ;TEST 17:       TEST THAT TRANSMIT READY BIT(5 OF CSR  SETS AND CLEARS
                ;               WHEN TCR IS SET AND CLEARED.
```

```
                                ;                ALSO CHECK THAT THE RIGHT LINE NO. (D) APPEARS IN RSLF.
                                ;PROBABLE FAULTY LOGIC:  MT295 (D2-5  ALL, (D2-6, E23, E32, E33, E49, (D2-8, E3, E5
                                ;*****************************************************************

001000  000400  175156  TST17:  MOV     #400,   @CSR    ;SET XMTR SCAN ENABLE
001004  000100  175154          BIS     #BIT6,  @TCR    ;SET XMTR CONTROL BIT, LINE 0
001010  000005                  RESET                   ;SET UP WAIT COUNTER
001012  175142          1$:     MOV     @CSR,   R1      ;CHECK AND SAVE XMTR READY
001016  100405                  BMI     2$              ;BRANCH IF SET OK
001020  005005                  CLR     R5              ;WAIT A WHILE
001022  001375                  BNE     .-4
001024  104401                  HLT+1                   ;XMTR READY FAILED TO SET
001026  000416                  BR      3$              ;SKIP LINE # CHECK ON ERROR
001030  012705  000100  2$:     MOV     #100,   R5      ;SET UP TIMER
001034  010104                  MOV     R1,     R4      ;SAVE LINE NUMBER
001036  042704  000377          BIC     #377,   R4      ;CHECK THAT THE XMTR STORED ON LINE 0
001042  001401                  BEQ     .+4             ;BRANCH IF OK
001044  104401                  HLT+1                   ;WRONG LINE NUMBER APPEARED IN RSLF
001046  175100          3$:     MOV     @CSR,   R2      ;CHECK AND SAVE XMTR READY
001052  100402                  BMI     .+6             ;BRANCH IF OK
001054  104402                  HLT+2                   ;READING THE RSLF CLEARED XMTR READY
001056  012700  175072  4$:     MOV     #TCR,   R0      ;CLR XMTR CONTROL BIT, LINE 0
001062  105010                  CLRB    (R0)            ;CHECK AND SAVE XMTR READY
001064  005005                  CLR     R5              ;BRANCH IF OK
001066  001375                  BNE     .-4             ;XMTR READY FAILED TO CLEAR WHEN
001070  104401                  HLT+1                   ;  XMTR CONTROL BIT WAS CLEARED

001072  104400                  SCOPE


                                ;*****************************************************************
                                ;TEST 20:       TEST THAT TRANSMIT READY BITS OF CSR  SETS AND CLEARS
                                ;                 WHEN EACH TCR BIT IS SET AND CLEARED
                                ;                ALSO CHECK THAT THE RIGHT LINE NO.  APPEARS IN RSLF.
                                ;PROBABLE FAULTY LOGIC:  MT295 (D2-5) ALL, (D2-6) E23, E32, E33, E49, (D2-8  E3, E5
                                ;*****************************************************************

001076  012737  000400  175046  TST20:  MOV     #400,   @CSR    ;SET XMTR SCAN ENABLE
001104  010100                  MOV     R1,     R0
001106  012701  000001          MOV     #1,     R1      ;SET UP MARKER
001112  012705  175040  4$:     MOV     #TCR,   R5      ;SET XMTR CONTROL BIT, EACH LINE
001116  105015                  CLRB    (R5)            ;SET UP WAIT COUNTER
001120  005005                  CLR     R5              ;CHECK AND SAVE XMTR READY
001122  175026          1$:     MOV     @CSR,   R2      ;BRANCH IF SET OK
001126  100405                  BMI     2$              ;WAIT A WHILE
001130  005005                  CLR     R5
001132  001375                  BNE     .-4
001134  104401                  HLT+1                   ;XMTR READY FAILED TO SET
001136  000416                  BR      3$              ;R1=LINE #
                                ;R2=CSR
```

```
                                                        ;SKIP LINE # CHECK ON ERROR
                                                        ;SET UP TIMER
                                                        ;SAVE LINE NUMBER
                                                        ;CHECK THAT THE XMTR STOPPED ON RIGHT LINE
                                                        ;BRANCH IF OK
                                                        ;WRONG LINE NUMBER APPEARED IN TBUF
                                                        ;R1=LINE #(SHOULD BE)
                                                        ;R2=LINE # (TBUF)

                                                        ;CHECK AND SAVE XMTR READY
                                                        ;BRANCH IF OK
                                                        ;READING THE TBUF CLEARED XMTR READY

                                                        ;CLR XMTR CONTROL BIT, EACH LINE
                                                        ;CHECK AND SAVE XMTR READY
                                                        ;BRANCH IF OK
                                                        ;XMTR READY FAILED TO CLEAR WHEN
                                                        ;    XMTR CONTROL FOR LINE -1 WAS CLEARED
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF CSR

                                                        ;INC LINE COUNTER TO NEXT LINE
                                                        ;SHIFT MARKER TO NEXT LINE
                                                        ;BRANCH IF NOT LAST LINE

                                                        SCOPE


;*****************************************************************************
;TEST 21:       TEST THAT MASTER TRANSMIT SCAN ENABLE (CSR BIT 8) ON A C
;               DISABLES TRANSMITTER READY (CSR BIT 15)
;               WHEN TOR BIT (LINE 0) IS SET.
;PROBABLE FAULTY LOGIC:  M7885 DZ-6  E49,E23,E32,E33
;*****************************************************************************

TST21:  CLR     @CSR                    ;CLEAR CONTROL STATUS
        BIS     #BIT0,@TCR              ;SET XMTR CONTROL BIT, LINE 0
        MOV     TIMER,R0                ;GET TIMER FROM PREVIOUS TEST
1$:     MOV     @CSR,R1                 ;CHECK AND SAVE XMTR READY
        BPL     2$                      ;BRANCH IF NOT SET
        HLT+1                           ;XMTR READY SET WITHOUT
        BR      3$                      ;MASTER TRANS SCAN ENABLE
2$:     DEC     R0                      ;WAIT A WHILE
        BNE     1$                      ;AND CHECK AGAIN

3$:     BIS     #400,@CSR               ;SET MASTER TRAN SCAN ENABLE
        CLR     R0
4$:     MOV     @CSR,   R1              ;CHECK AND SAVE XMTR READY
        BMI     5$
        INC     R0
        BNE     4$
        HLT+1                           ;TRAN READY NEVER CAME UP
                                        ;TOR BIT WAS SET FIRST
```

```
                                                        ;THEN MASTER TRAN SCAN ENABLE
004330  174656  5$:     BIC     #BIT0,  @TCR    ;CLEAR IT OUT
004334  105761  174646          MOV     @CSR,   R1      ;CHECK AND SAVE TRAN READY
004340  100401                  BPL     .+4             ;BRANCH IF OK
004342  104401                  HLT+1                   ;TRAN READY DIDN'T CLEAR.

004344  104422                  SCOPE


                        ;*****************************************************************
                        ;TEST 22:       TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 7
                        ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
                        ;*****************************************************************

004346  012737  004464  174646  TST22:  MOV     #ISR22,@XMT.EC  ;SET UP XMTR INTERRUPT VECTOR
004354  012737  000340  174642          MOV     #340,@XMTLVL    ;AT LEVEL 7
004362  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
004370  052737  000300  177776          BIS     #300,@#PS       ;SET PS TO LEVEL 6
004376  012777  040400  174460          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
004404  012777  000001  174460          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
004412  017701  174570          1$:     MOV     @CSR,   R1      ;WAIT
004416  100375                  BPL     1$
004420  000404                  BR      END22           ;Ok, BRANCH IF NO INTERRUPT

004422  104000          ISR22:  HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 7
004424  012716  004434          MOV     #END22,(SP)     ;MOVE NEW RTI ADR ONTO STACK
004430  000002                  RTI

004434  013777  001226  174562  END22:  MOV     XMTLVL,@XMTVEC
004442  012777  000340  174556          MOV     #IGT,   @XMTLVL
004450  005077  174540                  CLR     @TCR
004454  005077  174542                  CLR     @CSR
004460  042737  000340  177776          BIC     #340,@#PS

004466  104400                  SCOPE


                        ;*****************************************************************
                        ;TEST 23:       TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 6
                        ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
                        ;*****************************************************************

004470  012777  004544  174526  TST23:  MOV     #ISR23,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
004476  012777  000340  174522          MOV     #340,@XMTLVL    ;AT LEVEL 7
004504  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
004512  052737  000300  177776          BIS     #300,@#PS       ;SET PS TO LEVEL 6
004520  012777  040400  174462          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
004526  012777  000001  174460          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
004534  017701  174450          1$:     MOV     @CSR,   R1      ;WAIT
004540  100375                  BPL     1$
004542  000404                  BR      END23           ;OK, BRANCH IF NO INTERRUPT

004544  104000          ISR23:  HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 6
004546  012716  004564          MOV     #END23,(SP)     ;MOVE NEW RTI ADR ONTO STACK
004552  000002                  RTI
```

```
 1274  004554  013777  001226  174442  END23:  MOV     XMTLVL,@XMTVEC
 1275  004562  012777  000004  174436          MOV     #IOT,   @XMTLVL
 1276  004570  005077  174420                  CLR     @TCR
 1277  004574  005077  174410                  CLR     @CSR
 1278  004600  042737  000340  177776          BIC     #340,@#PS
 1279
 1280  004606  104400                          SCOPE
 1281
 1282
 1283
 1284                                  ;*****************************************************************
 1285                                  ;TEST 24:       TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 5
 1286                                  ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
 1287                                  ;*****************************************************************
 1288  004610  012777  004664  174406  TST24:  MOV     #ISR24,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
 1289  004616  012777  000340  174402          MOV     #340,@XMTLVL    ;AT LEVEL 7
 1290  004624  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
 1291  004632  052737  000240  177776          BIS     #240,@#PS       ;SET PS TO LEVEL 5
 1292  004640  012777  040400  174342          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
 1293  004646  012777  000001  174340          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
 1294  004654  017701  174330          1$:     MOV     @CSR,   R1      ;WAIT
 1295  004660  100375                          BPL     1$
 1296  004662  000404                          BR      END24           ;OK, BRANCH IF NO INTERRUPT
 1297
 1298  004664  104000                  ISR24:  HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 5
 1299  004666  012716  004674                  MOV     #END24,(SP)     ;MOVE NEW RTI ADR ONTO STACK
 1300  004672  000002                          RTI
 1301
 1302  004674  013777  001226  174322  END24:  MOV     XMTLVL,@XMTVEC
 1303  004702  012777  000004  174316          MOV     #IOT,   @XMTLVL
 1304  004710  005077  174300                  CLR     @TCR
 1305  004714  005077  174270                  CLR     @CSR
 1306  004720  042737  000340  177776          BIC     #340,@#PS
 1307
 1308  004726  104400                          SCOPE
 1309
 1310
 1311
 1312                                  ;*****************************************************************
 1313                                  ;TEST 25:       TEST THAT INTERRUPT OCCURS AT LEVEL 4
 1314                                  ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
 1315                                  ;*****************************************************************
 1316  004730  012777  005006  174266  TST25:  MOV     #ISR25,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
 1317  004736  012777  000340  174262          MOV     #340,@XMTLVL    ;AT LEVEL 7
 1318  004744  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
 1319  004752  052737  000200  177776          BIS     #200,@#PS       ;SET PS TO LEVEL 4
 1320  004760  012777  040400  174222          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
 1321  004766  012777  000001  174220          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
 1322  004774  017701  174210          1$:     MOV     @CSR,   R1      ;WAIT
 1323  005000  100375                          BPL     1$
 1324  005002  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 4
 1325  005004  000403                          BR      END25           ;CONTINUE
 1326
 1327  005006  012716  005014          ISR25:  MOV     #END25,(SP)     ;MOVE NEW RTI ADR ONTO STACK
 1328  005012  000002                          RTI
```

```
 1330  005014  013777  001226  174202  END25:  MOV     XMTLVL,@XMTVEC
 1331  005022  012777  000004  174176          MOV     #IOT,   @XMTLVL
 1332  005030  005077  174160                  CLR     @TCR
 1333  005034  005077  174150                  CLR     @CSR
 1334  005040  042737  000340  177776          BIC     #340,@#PS

       005046  104400                          SCOPE

                                       ;*************************************************************
                                       ;TEST 26:       TEST THAT INTERRUPT OCCURS AT LEVEL 3
                                       ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
                                       ;*************************************************************

       005050  013777  005126  174146  TST26:  MOV     #ISR26,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
       005056  012777  000340  174142          MOV     #340,@XMTLVL    ;AT LEVEL 7
       005064  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
       005072  052737  000140  177776          BIS     #140,@#PS       ;SET PS TO LEVEL 3
       005100  012777  040400  174102          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
       005106  012777  000001  174100          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
       005114  017701  174070          1$:     MOV     @CSR,   R1      ;WAIT
       005120  100375                          BPL     1$
       005122  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 3
       005124  000403                          BR      END26           ;CONTINUE

       005126  012716  005134          ISR26:  MOV     #END26,(SP)     ;MOVE NEW RTI ADR ONTO STACK
       005132  000002                          RTI

       005134  013777  001226  174062  END26:  MOV     XMTLVL,@XMTVEC
       005142  012777  000004  174056          MOV     #IOT,   @XMTLVL
       005150  005077  174046                  CLR     @TCR
       005154  005077  174036                  CLR     @CSR
       005160  042737  000340  177776          BIC     #340,@#PS

       005166  104400                          SCOPE

                                       ;*************************************************************
                                       ;TEST 27:       TEST THAT INTERRUPT OCCURS AT LEVEL 2
                                       ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
                                       ;*************************************************************

       005170  012777  005246  174026  TST27:  MOV     #ISR27,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
       005176  012777  000340  174022          MOV     #340,@XMTLVL    ;AT LEVEL 7
       005204  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
       005212  052737  000100  177776          BIS     #100,@#PS       ;SET PS TO LEVEL 2
       005220  012777  040400  173762          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
       005226  012777  000001  173760          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
       005234  017701  173750          1$:     MOV     @CSR,   R1      ;WAIT
       005240  100375                          BPL     1$
       005242  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 2
       005244  000403                          BR      END27           ;CONTINUE

       005246  012716  005254          ISR27:  MOV     #END27,(SP)     ;MOVE NEW RTI ADR ONTO STACK
       005252  000002                          RTI
```

```
                                                ;*****************************************************************
                                                ;TEST 30:       TEST THAT INTERRUPT OCCURS AT LEVEL 1
                                                ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
                                                ;*****************************************************************

005310  012777  005366  173706  TST30:  MOV     #ISR30,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
005316  012777  000340  173702           MOV     #340,@XMTLVL    ;AT LEVEL 7
005324  042737  000340  177776           BIC     #340,@#PS       ;CLEAR PS LEVEL
005332  052737  000040  177776           BIS     #040,@#PS       ;SET PS TO LEVEL 1
005340  012777  040400  173642           MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
005346  012777  000001  173640           MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
005354  017701  173630          1$:      MOV     @CSR,   R1      ;WAIT
005360  100375                          BPL     1$
005362  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 1
005364  000403                          BR      END30           ;CONTINUE

005366  012716  005374          ISR30:   MOV     #END30,(SP)     ;MOVE NEW RTI ADR ONTO STACK
005372  000002                          RTI

005374  013777  001226  173622  END30:   MOV     XMTLVL,@XMTVEC
005402  012777  000004  173616           MOV     #IOT,   @XMTLVL
005410  005077  173600                   CLR     @TCR
005414  005077  173570                   CLR     @CSR
005420  042737  000340  177776           BIC     #340,@#PS

005426  104400                          SCOPE


                                                ;*****************************************************************
                                                ;TEST 31:       TEST THAT INTERRUPT OCCURS AT LEVEL 0
                                                ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
                                                ;*****************************************************************

005430  012777  005506  173566  TST31:  MOV     #ISR31,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
005436  012777  000340  173562           MOV     #340,@XMTLVL    ;AT LEVEL 7
005444  042737  000340  177776           BIC     #340,@#PS       ;CLEAR PS LEVEL
005452  052737  000000  177776           BIS     #000,@#PS       ;SET PS TO LEVEL 0
005460  012777  040400  173522           MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
005466  012777  000001  173520           MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
005474  017701  173510          1$:      MOV     @CSR,   R1      ;WAIT
005500  100375                          BPL     1$
005502  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 0
005504  000403                          BR      END31           ;CONTINUE

005506  012716  005514          ISR31:   MOV     #END31,(SP)     ;MOVE NEW RTI ADR ONTO STACK
005512  000002                          RTI
```

```
 1442  005514  013777  001226  173502  END31:  MOV     XMTLVL,@XMTVEC
 1443  005522  012777  000004  173476          MOV     #IQT,   @XMTLVL
 1444  005530  105077  173460                  CLR     @TCR
 1445  005534  005077  173454                  CLR     @CSR
 1446  005540  042737  000340  177776          BIC     #340,@#PS
 1447
 1448  005546  104403                          SCOPE

 1449  005550  005037  001302                  CLR     TIMER           ;INITIALIZE TIMER
 1450  005554  012737  005562  015770          MOV     #.+6,   LAD     ;RESET LOOP ADDRESS

                                       ;*****************************************************************
                                       ;TEST 32:       TEST THAT LINE 0 CAN TRANSMIT AND
                                       ;               RECEIVE A CHARACTER. (377)
                                       ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                                       ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
                                       ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
                                       ;       4$:  CHECKS THAT RIGHT LINE # (0) IN FI/FO
                                       ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                                       ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                                       ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                                       ;       8$:  CHECKS THAT DONE CLEARS
                                       ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DD3 SERIES
                                       ;*****************************************************************

                                       ;               INITIALIZE
                                       ;       DEVICE"CSR"REGISTER
                                       ;                                               SET:
 1470  005562  004737  015566          TST32:  JSR     PC,     @#INITD :BIT2 = MAINTENANCE
                                                                       :BIT3 = CLEAR MOS
                                                                       :BIT8 = MASTER XMTR SCAN ENB
                                       ;WAIT FOR MOS TO CLEAR
                                       ;                               SET:
                                       ;                               :BIT0 = RECEIVER ENABLE
                                       ;
 1477  005566  052777  000001  173420          BIS     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE 0
 1478  005574  017701  173410          LOP32:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 1479  005600  100375                          BPL     LOP32
 1480  005602  012777  000377  173406          MOV     #377,   @TBUF   ;SEND A RUBOUT
 1481  005610  005000                          CLR     R0              ;CLEAR COUNTER
 1482  005612  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
 1483  005614  001376                          BNE     1$
 1484  005616  105777  173386                  TSTB    @CSR            ;WAIT FOR DONE
 1485  005622  100405                          BMI     2$              ;BRANCH WHEN DONE
 1486  005624  005200                          INC     R0              ;TIME COUNTER
 1487  005626  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
 1488  005630  017701  173354                  MOV     @CSR,   R1      ;SAVE CSR
 1489  005634  104001                          HLT+1                   ;DONE NEVER CAME UP
 1490                                                                   ;R1 = CONTENTS OF CSR
 1491  005636  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
 1492  005642  017701  173344                  MOV     @RBUF,  R1      ;READ THE FIFO
 1493  005646  100401                          BMI     .+4             ;BRANCH IF CHARACTER READY
 1494  005650  104001                          HLT+1                   ;CHARACTER READY DIDN'T SET
 1495                                                                   ;R1 = CONTENTS OF RBUF
 1496  005652  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
 1497  005656  001401                          BEQ     .+4             ;BRANCH IF NONE
```

```
1498  005660  104031                          HLT+1                   ;ERROR IN RECEIVED CHAR
1499                                                                  ;R1 = CONTENTS OF RBUF
1500                                                                  ;BIT14=UART OVERRUN
1501                                                                  ;BIT13=FRAMING ERROR
1502                                                                  ;BIT12=PARITY ERROR
1503  005662  010102                  4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
1504  005664  042702  170377                BIC     #170377,R2      ;MASK LINE#
1505  005670  000302                          SWAB    R2              ;LINE # IN LOW BYTE
1506  005672  122702  000000                CMPB    #0,     R2      ;CHECK LINE #
1507  005676  001401                          BEQ     .+4             ;BRANCH IF OK
1508  005700  104001                          HLT+1                   ;WRONG LINE # RECEIVED
1509                                                                  ;R1 = CONTENTS OF RBUF
1510                                                                  ;BITS8-11 = LINE #
1511  005702  117702  173400          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
1512  005706  130201                          BITB    R2,     R1      ;CHECK CHAR LENGTH.
1513  005710  001401                          BEQ     .+4             ;BRANCH IF OK
1514  005712  104002                          HLT+2                   ;WRONG CHARACTER LENGTH
1515                                                                  ;R1=DATA FROM FIFO
1516                                                                  ;R2=MASK (BITS SET NOT EXPECTED)
1517  005714  105102                  6$:     COMB    R2              ;REVERSE THE MASK
1518  005716  120102                          CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
1519  005720  001401                          BEQ     .+4             ;BRANCH IF OK
1520  005722  104002                          HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
1521                                                                  ;R1=DATA FROM FIFO (COMPLETE WORD)
1522                                                                  ;R2=DATA (LOW BYTE) EXPECTED
1523  005724  017701  173262          7$:     MOV     @RBUF,  R1      ;READ FIFO
1524  005730  100001                          BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
1525  005732  104001                          HLT+1                   ;CHARACTER PRESENT STAYED SET
1526                                                                  ;R1 = CONTENTS OF RBUF
1527  005734  017701  173250          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
1528  005740  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
1529  005744  001401                          BEQ     .+4             ;BRANCH IF OK
1530  005746  104001                          HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
1531                                                                  ;R1 = CONTENTS OF CSR
1532  005750  005077  173240                  CLR     @TCR            ;CLEAR TCR
1533  005754  005077  173230                  CLR     @CSR            ;CLEAR CSR
1534  005760  104400                          SCOPE
1535
1536                                  ;*****************************************************************
1537                                  ;TEST 33:       TEST THAT LINE 1 CAN TRANSMIT AND
1538                                  ;               RECEIVE A CHARACTER. (377)
1539                                  ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
1540                                  ;       2$:  CHECKS THAT CHAR PRESENT IS IN FIFO
1541                                  ;       3$:  CHECKS THAT NO ERRORS IN FIFO
1542                                  ;       4$:  CHECKS THAT RIGHT LINE # (1) IN FIFO
1543                                  ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
1544                                  ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
1545                                  ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
1546                                  ;       8$:  CHECKS THAT DONE CLEARS
1547                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DJ3 SERIES
1548                                  ;*****************************************************************
1549
1550                                  ;               INITIALIZE
1551                                  ;       DEVICE"CSR"REGISTER
1552                                  ;                               SET:
1553  005762  004737  015566  TST33:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
```

```
                                                                        ;BIT3 = CLEAR MOS
                                                                        ;BIT8 = MASTER XMTR SCAN ENB
                                        ;WAIT FOR MOS TO CLEAR
                                        ;                               SET:
                                        ;                               ;BIT0 = RECEIVER ENABLE
                                        ;
1560  005766  052777  000002  173320            BIS     #BIT1, @TCR     ;SET XMTR CONTROL BIT, LINE 1
1561  005774  017701  173210          LOP33:    MOV     @CSR, R1        ;WAIT FOR XMTR READY
1562  006000  100375                           BPL     LOP33
1563  005002  012777  000377  173206            MOV     #377., @TBUF    ;SEND A RUBOUT
1564  006010  005000                           CLR     R0              ;CLEAR COUNTER
1565  006012  105305                  1$:      DECB    R5              ;SHORT WAIT LOOP
1566  006014  001376                           BNE     1$
1567  006016  105777  173166                  TSTB    @CSR            ;WAIT FOR DONE
1568  006022  100405                           BMI     2$              ;BRANCH WHEN DONE
1569  006024  005200                           INC     R0              ;TIME COUNTER
1570  006026  001371                           BNE     1$              ;BRANCH IF NOT TIME-OUT
1571  006030  017701  173154                  MOV     @CSR, R1        ;SAVE CSR
1572  006034  104001                           HLT+1                   ;DONE NEVER CAME UP
1573                                                                    ;R1 = CONTENTS OF CSR
1574  006036  050037  001302          2$:      BIS     R0, TIMER       ;SAVE TIMER
1575  006042  017701  173144                  MOV     @RBUF, R1       ;READ THE FIFO
1576  006046  100401                           BMI     .+4             ;BRANCH IF CHARACTER READY
1577  006050  104001                           HLT+1                   ;CHARACTER READY DIDN'T SET
1578                                                                    ;R1 = CONTENTS OF RBUF
1579  006052  032701  070000          3$:      BIT     #70000, R1      ;CHECK FOR ERROR BITS
1580  006056  001401                           BEQ     .+4             ;BRANCH IF NONE
1581  006060  104001                           HLT+1                   ;ERROR IN RECEIVED CHAR
1582                                                                    ;R1 = CONTENTS OF RBUF
1583                                                                    ;BIT14=UART OVERRUN
1584                                                                    ;BIT13=FRAMING ERROR
1585                                                                    ;BIT12=PARITY ERROR
1586  006062  010102                  4$:      MOV     R1, R2          ;DUPLICATE DATA WORD
1587  006064  042702  170377                  BIC     #170377, R2     ;MASK LINE#
1588  006070  000302                           SWAB    R2              ;LINE # IN LOW BYTE
1589  006072  122702  000001                  CMPB    #1., R2         ;CHECK LINE #
1590  006076  001401                           BEQ     .+4             ;BRANCH IF OK
1591  006100  104001                           HLT+1                   ;WRONG LINE # RECEIVED
1592                                                                    ;R1 = CONTENTS OF RBUF
1593                                                                    ;BITS8-11 = LINE #
1594  006102  117702  173200          5$:      MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
1595  006106  130201                           BITB    R2, R1          ;CHECK CHAR LENGTH.
1596  006110  001401                           BEQ     .+4             ;BRANCH IF OK
1597  006112  104002                           HLT+2                   ;WRONG CHARACTER LENGTH
1598                                                                    ;R1=DATA FROM FIFO
1599                                                                    ;R2=MASK (BITS SET NOT EXPECTED)
1600  006114  105102                  6$:      COMB    R2              ;REVERSE THE MASK
1601  006116  120102                           CMPB    R1, R2          ;CHECK THE ACTURAL DATA
1602  006120  001401                           BEQ     .+4             ;BRANCH IF OK
1603  006122  104002                           HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
1604                                                                    ;R1=DATA FROM FIFO (COMPLETE WORD)
1605                                                                    ;R2=DATA (LOW BYTE) EXPECTED
1606  006124  017701  173062          7$:      MOV     @RBUF, R1       ;READ FIFO
1607  006130  100001                           BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
1608  006132  104001                           HLT+1                   ;CHARACTER PRESENT STAYED SET
1609                                                                    ;R1 = CONTENTS OF RBUF
```

```
 1610  006134  017701  173050          B$:     MOV     @CSR,   R1      ;SAVE THE CSR
 1611  006140  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
 1612  006144  001401                          BEQ     .+4             ;BRANCH IF OK
 1613  006146  104001                          HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
 1614                                                                  ;R1 = CONTENTS OF CSR
 1615  006150  005077  173040                  CLR     @TCR            ;CLEAR TCR
 1616  006154  005077  173030                  CLR     @CSR            ;CLEAR CSR
 1617  006160  104400                          SCOPE

                                ;*****************************************************************
                                ;TEST 34:       TEST THAT LINE 2 CAN TRANSMIT AND
                                ;               RECEIVE A CHARACTER. (377)
                                ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                                ;       2$:  CHECKS THAT CHAR PRESENT IS IN FIFO
                                ;       3$:  CHECKS THAT NO ERRORS IN FIFO
                                ;       4$:  CHECKS THAT RIGHT LINE # (2) IN FIFO
                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                                ;       8$:  CHECKS THAT DONE CLEARS
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DC3 SERIES
                                ;*****************************************************************

                                ;               INITIALIZE
                                ;       DEVICE"CSR"REGISTER
                                ;                                               SET:
 1636  006162  004737  015566  TST34:  JSR     PC,     @#INITD                 ;BITB = MAINTENANCE
                                                                                ;BIT3 = CLEAR MOS
                                                                                ;BIT8 = MASTER XMTR SCAN ENB
                                ;WAIT FOR MOS TO CLEAR
                                ;                                               SET:
                                ;                                               ;BIT0 = RECEIVER ENABLE
                                ;
 1643  006166  052777  000004  173020          BIS     #BIT2,  @TCR    ;SET XMTR CONTROL BIT, LINE 2
 1644  006174  017701  173010          LOP34:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 1645  006200  100375                          BPL     LOP34
 1646  006202  012777  000377  173006          MOV     #377,   @TBUF   ;SEND A RUBOUT
 1647  006210  005000                          CLR     R0              ;CLEAR COUNTER
 1648  006212  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
 1649  006214  001376                          BNE     1$
 1650  006216  105777  172766                  TSTB    @CSR            ;WAIT FOR DONE
 1651  006222  100405                          BMI     2$              ;BRANCH WHEN DONE
 1652  006224  005200                          INC     R0              ;TIME COUNTER
 1653  006226  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
 1654  006230  017701  172754                  MOV     @CSR,   R1      ;SAVE CSR
 1655  006234  104001                          HLT+1                   ;DONE NEVER CAME UP
                                                                        ;R1 = CONTENTS OF CSR
 1657  006236  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
 1658  006242  017701  172744                  MOV     @RBUF,  R1      ;READ THE FIFO
 1659  006246  100401                          BMI     .+4             ;BRANCH IF CHARACTER READY
 1660  006250  104001                          HLT+1                   ;CHARACTER READY DIDN'T SET
 1661                                                                  ;R1 = CONTENTS OF RBUF
 1662  006252  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
 1663  006256  001401                          BEQ     .+4             ;BRANCH IF NONE
 1664  006260  104001                          HLT+1                   ;ERROR IN RECEIVED CHAR
 1665                                                                  ;R1 = CONTENTS OF RBUF
```

```
                                                                ;BIT14=UART OVERRUN
                                                                ;BIT13=FRAMING ERROR
                                                                ;BIT12=PARITY ERROR
006262  010102                  4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
006264  042702  177377                  BIC     #177377,R2      ;MASK LINE#
006270  000302                          SWAB    R2              ;LINE # IN LOW BYTE
006272  122702  000002                  CMPB    #2,     R2      ;CHECK LINE #
006276  001401                          BEQ     .+4             ;BRANCH IF OK
006300  104001                          HLT+1                   ;WRONG LINE # RECEIVED
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BITS8-11 = LINE #
006302  117702  173000          5$:     MOVB    @DCLEN, R2      ;GET MASK OF CHARACTER
006306  130201                          BITB    R2,     R1      ;CHECK CHAR LENGTH.
006310  001401                          BEQ     .+4             ;BRANCH IF OK
006312  104002                          HLT+2                   ;WRONG CHARACTER LENGTH
                                                                ;R1=DATA FROM FIFO
                                                                ;R2=MASK (BITS SET NOT EXPECTED)
006314  105102                  6$:     COMB    R2              ;REVERSE THE MASK
006316  120102                          CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
006320  001401                          BEQ     .+4             ;BRANCH IF OK
006322  104002                          HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                                ;R1=DATA FROM FIFO (COMPLETE WORD)
                                                                ;R2=DATA (LOW BYTE) EXPECTED
006324  017701  172662          7$:     MOV     @RBUF,  R1      ;READ FIFO
006330  100001                          BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
006332  104001                          HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                                ;R1 = CONTENTS OF RBUF
006334  017701  172650          8$:     MOV     @RCSR,  R1      ;SAVE THE CSR
006340  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
006344  001401                          BEQ     .+4             ;BRANCH IF OK
006346  104001                          HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                                ;R1 = CONTENTS OF CSR
006350  005077  172640                  CLR     @TCR            ;CLEAR TCR
006354  005077  172630                  CLR     @RCSR           ;CLEAR CSR
006360  104400                          SCOPE

                                ;*****************************************************************
                                ;TEST 35:       TEST THAT LINE 3 CAN TRANSMIT AND
                                ;               RECEIVE A CHARACTER. (377)
                                ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                                ;       2$:  CHECKS THAT CHAR PRESENT IS IN FIFO
                                ;       3$:  CHECKS THAT NO ERRORS IN FIFO
                                ;       4$:  CHECKS THAT RIGHT LINE # (3) IN FIFO
                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                                ;       8$:  CHECKS THAT DONE CLEARS
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
                                ;*****************************************************************

                                ;               INITIALIZE
                                ;       DEVICE"CSR"REGISTER
                                ;                               SET:
006362  004737  015566          TST35:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT9 = MASTER XMTR SCAN ENB
```

```
                                        ;WAIT FOR MOS TO CLEAR
                                        ;                                       SET:
                                        ;                                       ;BIT0 = RECEIVER ENABLE
                                        ;
006366  052777  000010  172620          BIS     #BIT3,  @TCR    ;SET XMTR CONTROL BIT, LINE 3
006374  017701  172610          LOP35:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
006400  100375                  BPL     LOP35
006402  012777  000377  172606          MOV     #377,   @TBUF   ;SEND A RUBOUT
006410  005000                  CLR     R0              ;CLEAR COUNTER
006412  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
006414  001376                  BNE     1$
006416  105777  172566                  TSTB    @CSR            ;WAIT FOR DONE
006422  100405                  BMI     2$              ;BRANCH WHEN DONE
006424  005200                  INC     R0              ;TIME COUNTER
006426  001371                  BNE     1$              ;BRANCH IF NOT TIME-OUT
006430  017701  172554                  MOV     @CSR,   R1      ;SAVE CSR
006434  104001                  HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
006436  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
006442  017701  172544                  MOV     @RBUF,  R1      ;READ THE FIFO
006446  100401                  BMI     .+4             ;BRANCH IF CHARACTER READY
006450  104001                  HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
006452  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
006456  001401                  BEQ     .+4             ;BRANCH IF NONE
006460  104001                  HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR
006462  010102          4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
006464  042702  170377          BIC     #170377,R2      ;MASK LINE#
006470  000302                  SWAB    R2              ;LINE # IN LOW BYTE
006472  122702  000003          CMPB    #3.,    R2      ;CHECK LINE #
006476  001401                  BEQ     .+4             ;BRANCH IF OK
006500  104001                  HLT+1                   ;WRONG LINE # RECEIVED
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 => LINE #
006502  117702  172600          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
006506  130201                  BITB    R2,     R1      ;CHECK CHAR LENGTH.
006510  001401                  BEQ     .+4             ;BRANCH IF OK
006512  104002                  HLT+2                   ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FIFO
                                                        ;R2=MASK (BITS SET NOT EXPECTED)
006514  105102          6$:     COMB    R2              ;REVERSE THE MASK
006516  120102                  CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
006520  001401                  BEQ     .+4             ;BRANCH IF OK
006522  104002                  HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FIFO (COMPLETE WORD)
                                                        ;R2=DATA (LOW BYTE) EXPECTED
006524  017701  172462          7$:     MOV     @RBUF,  R1      ;READ FIFO
006530  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
006532  104001                  HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                        ;R1 = CONTENTS OF RBUF
006534  017701  172450          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
006540  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
```

```
[illegible]                         BEQ     .+4             ;BRANCH IF OK
[illegible]                         HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                            ;R1 = CONTENTS OF CSR
[illegible]                         CLR     @TCR            ;CLEAR TCR
[illegible]                         CLR     @CSR            ;CLEAR CSR
[illegible]                         SCOPE
[illegible]                         MOV     DCLEN
[illegible]                         MOV     #.+6,   LAD

                        ;*****************************************************************
                        ;TEST 36:       TEST THAT LINE 4 CAN TRANSMIT AND
                        ;               RECEIVE A CHARACTER. (377)
                        ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                        ;       2$:  CHECKS THAT CHAR PRESENT IS IN FIFO
                        ;       3$:  CHECKS THAT NO ERRORS IN FIFO
                        ;       4$:  CHECKS THAT RIGHT LINE # (4) IN FIFO
                        ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                        ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                        ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                        ;       8$:  CHECKS THAT DONE CLEARS
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CHIP DC3 SERIES
                        ;*****************************************************************

                        ;               INITIALIZE
                        ;       DEVICE CSR REGISTER
                        ;                                                   SET:
 [illegible]            TST36:  JSR     PC,     @#INITD     ;BIT3 = MAINTENANCE
                                                            ;BIT4 = CLEAR MDS
                                                            ;BIT5 = MASTER XMTR SCAN ENB

                        ;WAIT FOR MDS TO CLEAR
                        ;                                                   SET:
                        ;                                                   ;BIT12 = RECEIVER ENABLE
                        ;
 [illegible]  172405            BIS     #BIT4,  @TCR        ;SET XMTR CONTROL BIT, LINE 4
 [illegible]            LOP36:  [illegible]     R1          ;WAIT FOR XMTR READY
 [illegible]                    BPL     [illegible]
 [illegible]  000377  172374    MOVB    #377,   @TBUF       ;SEND A RUBOUT
 [illegible]                    CLR     R0                  ;CLEAR COUNTER
 [illegible]            1$:     DECB    R5                  ;SHORT WAIT LOOP
 [illegible]                    BNE     1$
 [illegible]  172354            TSTB    @CSR                ;WAIT FOR DONE
 [illegible]                    BMI     2$                  ;BRANCH WHEN DONE
 [illegible]                    INC     R0                  ;TIME COUNTER
 [illegible]                    BNE     1$                  ;BRANCH IF NOT TIME-OUT
 [illegible]  172342            MOV     @CSR,   R1          ;SAVE CSR
 [illegible]                    HLT+1                       ;DONE NEVER CAME UP
                                                            ;R1 = CONTENTS OF CSR
 [illegible]            2$:     MOV     R0,     TIMER       ;SAVE TIMER
 [illegible]                    MOV     @RBUF,  R1          ;READ THE FIFO
 [illegible]                    BMI     .+4                 ;BRANCH IF CHARACTER READY
 [illegible]                    HLT+1                       ;CHARACTER READY DIDN'T SET
                                                            ;R1 = CONTENTS OF RBUF
 [illegible]  070000    3$:     BIT     #70000, R1          ;CHECK FOR ERROR BITS
 [illegible]                    BEQ     .+4                 ;BRANCH IF NONE
 [illegible]                    HLT+1                       ;ERROR IN RECEIVED CHAR
                                                            ;R1 = CONTENTS OF RBUF
```

```
                                ;TEST LINE 4 TRANSMIT AND RECEIVE LOGIC

                                ;BIT14=UART OVERRUN
                                ;BIT13=FRAMING ERROR
                                ;BIT12=PARITY ERROR
                                ;DUPLICATE DATA WORD
                                ;MASK LINE#
                                ;LINE # IN LOW BYTE
                                ;CHECK LINE #
                                ;BRANCH IF OK
                                ;WRONG LINE # RECEIVED
                                ;R1 = CONTENTS OF RBUF
                                ;BITS8-11 = LINE #
                                ;GET MASK OF CHARACTER
                                ;CHECK CHAR LENGTH.
                                ;BRANCH IF OK
                                ;WRONG CHARACTER LENGTH
                                ;R1=DATA FROM FIFO
                                ;R2=MASK (BITS SET NOT EXPECTED
                                ;REVERSE THE MASK
                                ;CHECK THE ACTURAL DATA
                                ;BRANCH IF OK
                                ;WRONG CHAR LEN OR DATA ERROR
                                ;R1=DATA FROM FIFO (COMPLETE WORD.
                                ;R2=DATA (LOW BYTE) EXPECTED
                                ;READ FIFO
                                ;BRANCH IF CHAR PRESENT NOT SET
                                ;CHARACTER PRESENT STAYED SET
                                ;R1 = CONTENTS OF RBUF
                                ;SAVE THE CSR
                                ;CLEAR THE CSR
                                ;BRANCH IF OK
                                ;SOME OTHER CLEAR OR OTHER CSR ERROR
                                ;R1 = CONTENTS OF CSR
                                ;CLEAR
                                ;CLEAR CSR

;*****************************************************************
;
;       TEST 37:        TEST THE LINE 5 CHAR TRANSMIT AND
;                       RECEIVE A CHARACTER.
;
;               CHECK THAT DONE SETS IN RECEIVE SIDE.
;               CHECK THAT DONE IS PRESENT
;               CHECK FOR NO ERRORS IN
;               CHECK THAT RIGHT LINE #
;               CHECK FOR RIGHT CHARACTER
;               CHECK THAT CORRECT DATA
;               CHECK THAT CHARACTER PRESENT
;               CHECK THAT DONE CLEARS
;
;*****************************************************************

;
;       INITIALIZE
;       DEVICE CSR REGISTER
;
                                ;SET:
                                ;       = MAINTENANCE
                                ;       = [illegible]
                                ;       = [illegible]
```

```
                                        ;WAIT FOR XMB TO CLEAR
                                        ;
                                        ;
                                                        SET:
                                                        ;BIT0 = RECEIVER ENABLE

000340  172526          LOP3$:  BIS     #BIT5,  @TCR    ;SET XMTR CONTROL BIT, LINE 5
172176                  1$:     TSTB    @TCSR,  R1      ;WAIT FOR XMTR READY
                                BPL     LOP3$
000377  172174                  MOV     #377,   @TBUF   ;SEND A RUBOUT
                                CLR     R5              ;CLEAR COUNTER
                        18:     INCB    R5              ;SHORT WAIT LOOP
                                BNE     1$

172154                          TSTB    @RCSR           ;WAIT FOR DONE
                                BPL     .+4             ;BRANCH WHEN DONE
                                INC     R5              ;TIME COUNTER
                                BNE     1$              ;BRANCH IF NOT TIME-OUT

172142                          MOV     @RCSR,  R1      ;SAVE CSR
                                HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR

000200          23:             MOV     R0,     -(SP)   ;SAVE TIMER
172136                          MOV     @RBUF,  R1      ;READ THE FIFO
                                BMI     .+4             ;BRANCH IF CHARACTER READY
                                HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF

070000          33:             BIT     #70000, R1      ;CHECK FOR ERROR BITS
                                BEQ     .+4             ;BRANCH IF NONE
                                HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR

                        43:     MOV     R1,     R2      ;DUPLICATE DATA WORD
177377                          BIC     #177377,R2      ;MASK LINE#
                                SWAB    R2              ;LINE # IN LOW BYTE
000005                          CMPB    #5.,    R2      ;CHECK LINE #
                                BEQ     .+4             ;BRANCH IF OK
                                HLT+2                   ;WRONG LINE # RECEIVED
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT5-11 = LINE #

172166          53:             MOVB    $DULEN, R2      ;GET MASK OF CHARACTER
                                CMPB    R2,     R1      ;CHECK CHAR LENGTH.
                                BEQ     .+4             ;BRANCH IF OK
                                HLT+2                   ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FIFO
                                                        ;R2=MASK (BITS SET NOT EXPECTED)

                        63:     COMB    R2              ;REVERSE THE MASK
                                BICB    R1,     R2      ;CHECK THE ACTURAL DATA
                                BEQ     .+4             ;BRANCH IF OK
                                HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FIFO (COMPLETE WORD)
                                                        ;R2=DATA (LOW BYTE) EXPECTED

172050          73:             MOV     @RBUF,  R1      ;READ FIFO
                                BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
                                HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                        ;R1 = CONTENTS OF RBUF

172036          83:             MOV     @CSR,   R1      ;SAVE THE CSR
100405                          CMP     #100405,R1      ;CHECK THE CSR
```

```
                                                BEQ     .+4             ;BRANCH IF OK
                                                HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                                        ;R1 = CONTENTS OF CSR
                                                CLR     @TCR            ;CLEAR TCR
                                                CLR     @CSR            ;CLEAR CSR
                                                SCOPE

                                ;*****************************************************************************
                                ;TEST 40:       TEST THAT LINE 6 CAN TRANSMIT AND
                                ;               RECEIVE A CHARACTER. (377)
                                ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                                ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
                                ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
                                ;       4$:  CHECKS THAT RIGHT LINE # (6) IN FI/FO
                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                                ;       8$:  CHECKS THAT DONE CLEARS
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-?, ALL; M7279 ALL; UART CARD DD3 SERIES
                                ;*****************************************************************************

                                ;               INITIALIZE
                                ;       DEVICE"CSR"REGISTER
                                                                        SET:
TST40:  JSP     PC,     @#INITD                 ;BIT2 = MAINTENANCE
                                                                        ;BIT3 = CLEAR MOS
                                                                        ;BIT8 = MASTER XMTR SCAN ENB

                                ;WAIT FOR MOS TO CLEAR
                                ;
                                ;
                                ;
                                                                        SET:
                                                                        ;BIT0 = RECEIVER ENABLE

        BIS     #BIT6,  @TCR    ;SET XMTR CONTROL BIT, LINE 6
LOP40:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
        BPL     LOP40
        MOV     #377,   @TBUF   ;SEND A RUBOUT
        CLR     R0              ;CLEAR COUNTER
        DECB    R5              ;SHORT WAIT LOOP
1$:     BNE     1$
        TSTB    @CSR            ;WAIT FOR DONE
        BMI     2$              ;BRANCH WHEN DONE
        INC     R0              ;TIME COUNTER
        BNE     1$              ;BRANCH IF NOT TIME-OUT
        MOV     @CSR,   R1      ;SAVE CSR
        HLT+1                   ;DONE NEVER CAME UP
                                ;R1 = CONTENTS OF CSR
2$:     BIS     R0,     TIMER   ;SAVE TIMER
        MOV     @RBUF,  R1      ;READ THE FI FO
        BMI     .+4             ;BRANCH IF CHARACTER READY
        HLT+1                   ;CHARACTER READY DIDN'T SET
                                ;R1 = CONTENTS OF RBUF
3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
        BEQ     .+4             ;BRANCH IF NONE
        HLT+1                   ;ERROR IN RECEIVED CHAR
                                ;R1 = CONTENTS OF RBUF
                                ;BIT14=UART OVERRUN
                                ;BIT13=FRAMING ERROR
```

```
                                                        ;BIT12=PARITY ERROR
                                                        ;DUPLICATE DATA WORD
007332  013102          4$:     MOV     R1,     R2      ;MASK LINE#
007334  042702  170377          BIC     #170377,R2      ;LINE # IN LOW BYTE
007340  000302                  SWAB    R2              ;CHECK LINE #
007342  122702  000006          CMPB    #6.,    R2      ;BRANCH IF OK
007346  001401                  BEQ     .+4             ;WRONG LINE # RECEIVED
007350  104001                  HLT+1                   ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 = LINE #
007352  117702  171756          5$:     MOVB    @DILEN, R2      ;GET MASK OF CHARACTER
007356  130201                  BITB    R2,     R1      ;CHECK CHAR LENGTH.
007360  001401                  BEQ     .+4             ;BRANCH IF OK
007362  104002                  HLT+2                   ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FI/FO
                                                        ;R2=MASK (BITS SET NOT EXPECTED
007364  105102          6$:     COMB    R2              ;REVERSE THE MASK
007366  130102                  CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
007370  001401                  BEQ     .+4             ;BRANCH IF OK
007372  104003                  HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                        ;R2=DATA (LOW BYTE) EXPECTED
007374  017701  171650          7$:     MOV     @RBUF,  R1      ;READ FI/FO
007400  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
007402  104001                  HLT+1                   ;CHARACTER PRESENT STAYED SE
                                                        ;R1 = CONTENTS OF RBUF
007404  017701  171636          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
007410  022701  100405          CMP     #100405,R1      ;CHECK THE CSR
007414  001401                  BEQ     .+4             ;BRANCH IF OK
007416  104001                  HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                        ;R1 = CONTENTS OF CSR
007420  005077  171626          CLR     @TCR            ;CLEAR TCR
007424  005077  171616          CLR     @CSR            ;CLEAR CSR
007430  104400                  SCOPE

                        ;*****************************************************************
                        ;TEST 41:       TEST THAT LINE 7 CAN TRANSMIT AND
                        ;               RECEIVE A CHARACTER. (377)
                        ;
                        ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                        ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI-FO
                        ;       3$:  CHECKS THAT NO ERRORS IN FI-FO
                        ;       4$:  CHECKS THAT RIGHT LINE # (7) IN FI-FO
                        ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                        ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                        ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                        ;       8$:  CHECKS THAT DONE CLEARS
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
                        ;*****************************************************************

                        ;               INITIALIZE
                        ;       DEVICE"CSR"REGISTER
                        ;                                               SET:
007432  004737  015566  TST41:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
                        ;WAIT FOR MOS TO CLEAR
                        ;                               SET:
```

```
                                                        ;
                                                        ;
                                                        ;BIT0 = RECEIVER ENABLE
                                                        ;
000000 000414 052737  000200  171636          BIS     #BIT7,@#TCR     ;SET XMTR CONTROL BIT, LINE 7
000004 000420 013701  171636          LOP41:  MOV     @#CSR,R1        ;WAIT FOR XMTR READY
000010 000424 100375                  BPL     LOP41
000012 000426 012737  000377  171634          MOV     #377,@#TBUF     ;SEND A RUBOUT
000020 000434 005000                  CLR     R0                      ;CLEAR COUNTER
000022 000436 005300                  1$:     DECB    R0              ;SHORT WAIT LOOP
000024 000440 001376                  BNE     1$
000026 000442 105737  171554          TSTB    @#CSR                   ;WAIT FOR DONE
000032 000446 100004                  BMI     2$                      ;BRANCH WHEN DONE
000034 000450 005200                  INC     R0                      ;TIME COUNTER
000036 000452 001372                  BNE     1$                      ;BRANCH IF NOT TIME-OUT
000040 000454 013701  171546          MOV     @#CSR,R1                ;SAVE CSR
000044 000460 104001                  HLT+1                           ;DONE NEVER CAME UP
                                                                      ;R1 = CONTENTS OF CSR
000046 000462 050037  001300          2$:     BIS     R0,TIMER        ;SAVE TIMER
000052 000466 013701  171630          MOV     @#RBUF,R1               ;READ THE FIFO
000056 000472 100401                  BMI     .+4                     ;BRANCH IF CHARACTER READY
000060 000474 104001                  HLT+1                           ;CHARACTER READY DIDN'T SET
                                                                      ;R1 = CONTENTS OF RBUF
000062 000476 032737  070000          3$:     BIT     #70000,R1       ;CHECK FOR ERROR BITS
000070 000504 001401                  BEQ     .+4                     ;BRANCH IF NONE
000072 000506 104001                  HLT+1                           ;ERROR IN RECEIVED CHAR
                                                                      ;R1 = CONTENTS OF RBUF
                                                                      ;BIT14=UART OVERRUN
                                                                      ;BIT13=FRAMING ERROR
                                                                      ;BIT12=PARITY ERROR
000074 000510 010104                  4$:     MOV     R1,R4           ;DUPLICATE DATA WORD
000076 000512 042704  177377          BIC     #177377,R4              ;MASK LINE#
000102 000516 020427  000007          CMP     R4,#7                   ;LINE # IN LOW BYTE
000106 000522 001401                  BEQ     .+4                     ;CHECK LINE #
000110 000524 104001                  HLT+1                           ;BRANCH IF OK
                                                                      ;WRONG LINE # RECEIVED
                                                                      ;R1 = CONTENTS OF RBUF
                                                                      ;BITS0-11 = LINE #
000112 000526 112737  171566  171566  5$:     MOVB    @RCVLEN,@RCVLEN ;GET MASK OF CHARACTER
000120 000534 001401                  BEQ     .+4                     ;CHECK CHAR LENGTH.
000122 000536 104002                  HLT+2                           ;BRANCH IF OK
                                                                      ;WRONG CHARACTER LENGTH
                                                                      ;R1=DATA FROM FIFO
                                                                      ;R2=MASK (BITS SET NOT EXPECTED
000124 000540 005102                  6$:     COM     R2              ;REVERSE THE MASK
000126 000542 030102                  BIT     R1,R2                   ;CHECK THE ACTURAL DATA
000130 000544 001401                  BEQ     .+4                     ;BRANCH IF OK
000132 000546 104002                  HLT+2                           ;WRONG CHAR LEN OR DATA ERROR
                                                                      ;R1=DATA FROM FIFO (COMPLETE WORD
                                                                      ;R2=DATA (LOW BYTE) EXPECTED
000134 000550 017701  171450          7$:     MOV     @RBUF,R1        ;READ FIFO
000140 000554 100401                  BPL     .+4                     ;BRANCH IF CHAR PRESENT NOT SET
000142 000556 104001                  HLT+1                           ;CHARACTER PRESENT STAYED SET
                                                                      ;R1 = CONTENTS OF RBUF
000144 000560 017701  171436          8$:     MOV     @CSR,R1         ;SAVE THE CSR
000150 000564 020127  100405          CMP     R1,#100405              ;CHECK THE CSR
000154 000570 001401                  BEQ     .+4                     ;BRANCH IF OK
000156 000572 104001                  HLT+1                           ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
```

```
                                                                        ;R1 = CONTENTS OF CSR
                                                CLR     @TCR            ;CLEAR TCR
                                                CLR     @CSR            ;CLEAR CSR
                                                SCOPE
                                                INC     DJLEN
                                                MOV     #.+6,   LAD

                                ;************************************************************
                                ;TEST 42:       TEST THAT LINE 8 CAN TRANSMIT AND
                                ;               RECEIVE A CHARACTER. (377)
                                ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                                ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
                                ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
                                ;       4$:  CHECKS THAT RIGHT LINE # (8) IN FI/FO
                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                                ;       8$:  CHECKS THAT DONE CLEARS
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DD3 SERIES
                                ;************************************************************

                                ;               INITIALIZE
                                ;       DEVICE CSR REGISTER
                                                                        ;SET:
007606  004737  016666          TST42:  JSR     PC,     @#INITD         ;BIT2 = MAINTENANCE
                                                                        ;BIT3 = CLEAR MOS
                                                                        ;BIT8 = MASTER XMTR SCAN ENB
                                ;WAIT FOR MOS TO CLEAR
                                ;
                                ;
                                ;
                                                                        ;SET:
                                                                        ;BIT0 = RECEIVER ENABLE

007612  052777  000400  171374          BIS     #BIT8,  @TCR    ;SET XMTR CONTROL BIT, LINE 8
007620  017701  171364          LOP42:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
007624  100375                  BPL     LOP42
007626  012777  000377  171362          MOV     #377,   @TBUF   ;SEND A RUBOUT
007634  005000                  CLR     R0              ;CLEAR COUNTER
007636  105305          1$:     DECB    R5              ;SHORT WAIT LOOP
007640  001376                  BNE     1$
007642  105777  171342          TSTB    @CSR            ;WAIT FOR DONE
007646  100405                  BMI     2$              ;BRANCH WHEN DONE
007650  005200                  INC     R0              ;TIME COUNTER
007652  001371                  BNE     1$              ;BRANCH IF NOT TIME-OUT
007654  017701  171330          MOV     @CSR,   R1      ;SAVE CSR
007660  104001                  HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
007662  050037  001332          2$:     BIS     R0,     TIMER   ;SAVE TIMER
007666  017701  171320          MOV     @RBUF,  R1      ;READ THE FI/FO
007672  100401                  BMI     .+4             ;BRANCH IF CHARACTER READY
007674  104001                  HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
007676  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
007702  001401                  BEQ     .+4             ;BRANCH IF NONE
007704  104001                  HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
```

```
                                                        ;BIT12=PARITY ERROR
                                                        ;DUPLICATE DATA WORD
007706  010102                    4$:     MOV     R1,     R2      ;MASK LINE#
007710  042702  170377                    BIC     #^C377,R2       ;LINE # IN LOW BYTE
007714  000302                            SWAB    R2              ;CHECK LINE #
007716  122702  000010                    CMPB    #8.,    R2      ;BRANCH IF OK
007722  001401                            BEQ     .+4             ;WRONG LINE # RECEIVED
007724  104001                            HLT+1                   ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 = LINE #
007726  117702  171354                    5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
007732  130201                            BITB    R2,     R1      ;CHECK CHAR LENGTH.
007734  001401                            BEQ     .+4             ;BRANCH IF OK
007736  104002                            HLT+2                   ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FIFO
                                                        ;R2=MASK (BITS SET NOT EXPECTED
007740  105102                    6$:     COMB    R2              ;REVERSE THE MASK
007742  120102                            CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
007744  001401                            BEQ     .+4             ;BRANCH IF OK
007746  104002                            HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FIFO (COMPLETE WORD
                                                        ;R2=DATA (LOW BYTE) EXPECTED
007750  017701  171235                    7$:     MOV     @RBUF,  R1      ;READ FIFO
007754  100001                            BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
007756  104001                            HLT+1                   ;CHARACTER PRESENT STAYED SE
                                                        ;R1 = CONTENTS OF RBUF
007760  017701  171224                    8$:     MOV     @CSR,   R1      ;SAVE THE CSR
007764  022701  100405                    CMP     #100405,R1      ;CHECK THE CSR
007770  001401                            BEQ     .+4             ;BRANCH IF OK
007772  104001                            HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                        ;R1 = CONTENTS OF CSR
007774  005077  171214                    CLR     @TCR            ;CLEAR TCR
010000  005077  171204                    CLR     @CSR            ;CLEAR CSR
010004  104400                            SCOPE

                                ;*****************************************************************
                                ;TEST 43:       TEST THAT LINE 9 CAN TRANSMIT AND
                                ;               RECEIVE A CHARACTER. (377)
                                ;
                                ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                                ;       2$:  CHECKS THAT CHAR PRESENT IS IN FIFO
                                ;       3$:  CHECKS THAT NO ERRORS IN FIFO
                                ;       4$:  CHECKS THAT RIGHT LINE # (9) IN FIFO
                                ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                                ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                                ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                                ;       8$:  CHECKS THAT DONE CLEARS
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DD3 SERIES
                                ;*****************************************************************

                                ;               INITIALIZE
                                ;       DEVICE"CSR"REGISTER
                                ;                                               SET:
010006  004737  015566          TST43:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT8 = MASTER XMTR SCAN ENB
                                ;WAIT FOR MOS TO CLEAR
                                ;                                               SET:
```

```
                                                    ;
                                                    ;                                   ;BIT0 = RECEIVER ENABLE
                                                    ;
010012  052777  001000  171174          BIS     #BIT9,  @TCR    ;SET XMTR CONTROL BIT, LINE 9
010020  017701  171164          LOP43:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
010024  100375                  BPL     LOP43
010026  012777  000377  171162          MOV     #377,   @TBUF   ;SEND A RUBOUT
010034  005000                  CLR     R0                      ;CLEAR COUNTER
010036  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
010040  001376                  BNE     1$
010042  105777  171142                  TSTB    @CSR            ;WAIT FOR DONE
010046  100405                  BMI     2$                      ;BRANCH WHEN DONE
010050  005200                  INC     R0                      ;TIME COUNTER
010052  001371                  BNE     1$                      ;BRANCH IF NOT TIME-OUT
010054  017701  171130                  MOV     @CSR,   R1      ;SAVE CSR
010060  104001                  HLT+1                           ;DONE NEVER CAME UP
                                                                ;R1 = CONTENTS OF CSR
010062  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
010066  017701  171120                  MOV     @RBUF,  R1      ;READ THE FIFO
010072  100401                  BMI     .+4                     ;BRANCH IF CHARACTER READY
010074  104001                  HLT+1                           ;CHARACTER READY DIDN'T SET
                                                                ;R1 = CONTENTS OF RBUF
010076  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
010102  001401                  BEQ     .+4                     ;BRANCH IF NONE
010104  104001                  HLT+1                           ;ERROR IN RECEIVED CHAR
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BIT14=UART OVERRUN
                                                                ;BIT13=FRAMING ERROR
                                                                ;BIT12=PARITY ERROR
010106  010102          4$:     MOV     R1,     R2              ;DUPLICATE DATA WORD
010110  042702  170377          BIC     #170377,R2              ;MASK LINE#
010114  000302                  SWAB    R2                      ;LINE # IN LOW BYTE
010116  122702  000011          CMPB    #9.,    R2              ;CHECK LINE #
010122  001401                  BEQ     .+4                     ;BRANCH IF OK
010124  104001                  HLT+1                           ;WRONG LINE # RECEIVED
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BITS8-11 = LINE #
010126  117702  171154          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
010132  130201                  BITB    R2,     R1              ;CHECK CHAR LENGTH.
010134  001401                  BEQ     .+4                     ;BRANCH IF OK
010136  104002                  HLT+2                           ;WRONG CHARACTER LENGTH
                                                                ;R1=DATA FROM FIFO
                                                                ;R2=MASK (BITS SET NOT EXPECTED)
010140  105102          6$:     COMB    R2                      ;REVERSE THE MASK
010142  120102                  CMPB    R1,     R2              ;CHECK THE ACTURAL DATA
010144  001401                  BEQ     .+4                     ;BRANCH IF OK
010146  104002                  HLT+2                           ;WRONG CHAR LEN OR DATA ERROR
                                                                ;R1=DATA FROM FIFO (COMPLETE WORD)
                                                                ;R2=DATA (LOW BYTE) EXPECTED
010150  017701  171036          7$:     MOV     @RBUF,  R1      ;READ FIFO
010154  100001                  BPL     .+4                     ;BRANCH IF CHAR PRESENT NOT SET
010156  104001                  HLT+1                           ;CHARACTER PRESENT STAYED SET
                                                                ;R1 = CONTENTS OF RBUF
010160  017701  171024          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
010164  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
010170  001401                  BEQ     .+4                     ;BRANCH IF OK
010172  104001                  HLT+1                           ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
```

```
                                                        ;R1 = CONTENTS OF CSR
010174  005077  171014                 CLR     @TCR     ;CLEAR TCR
010200  005077  171004                 CLR     @CSR     ;CLEAR CSR
010204  104400                         SCOPE

                              ;*****************************************************************
                              ;TEST 44:       TEST THAT LINE 10 CAN TRANSMIT AND
                              ;               RECEIVE A CHARACTER. (377)
                              ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                              ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
                              ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
                              ;       4$:  CHECKS THAT RIGHT LINE # (10) IN FI/FO
                              ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                              ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                              ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                              ;       8$:  CHECKS THAT DONE CLEARS
                              ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DC3 SERIES
                              ;*****************************************************************

                              ;               INITIALIZE
                              ;       DEVICE"CSR"REGISTER
                                                        ;SET:
010206  004737  015566        TST44:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
                              ;WAIT FOR MOS TO CLEAR
                              ;
                                                        ;SET:
                              ;
                                                        ;BIT0 = RECEIVER ENABLE
                              ;
010212  052777  002000  170774         BIS     #BIT10, @TCR    ;SET XMTR CONTROL BIT, LINE 10
010220  017701  170764         LOP44:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
010224  100375                 BPL     LOP44
010226  012777  000377  170762         MOV     #377,   @TBUF   ;SEND A RUBOUT
010234  005000                 CLR     R0              ;CLEAR COUNTER
010236  105305                 1$:     DECB    R5              ;SHORT WAIT LOOP
010240  001376                 BNE     1$
010242  105777  170742                 TSTB    @CSR            ;WAIT FOR DONE
010246  100405                 BMI     2$              ;BRANCH WHEN DONE
010250  005200                 INC     R0              ;TIME COUNTER
010252  001371                 BNE     1$              ;BRANCH IF NOT TIME-OUT
010254  017701  170730                 MOV     @CSR,   R1      ;SAVE CSR
010260  104001                 HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
010262  050037  001302         2$:     BIS     R0,     TIMER   ;SAVE TIMER
010266  017701  170720                 MOV     @RBUF,  R1      ;READ THE FI/FO
010272  100401                 BMI     .+4             ;BRANCH IF CHARACTER READY
010274  104001                 HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
010276  032701  070000         3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
010302  001401                 BEQ     .+4             ;BRANCH IF NONE
010304  104001                 HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR
010306  010102                 4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
```

```
 2338  010310  042702  170377          BIC     #170377,R2      ;MASK LINE#
 2339  010314  000302                  SWAB    R2              ;LINE # IN LOW BYTE
 2340  010316  122702  000012          CMPB    #10.,   R2      ;CHECK LINE #
 2341  010322  001401                  BEQ     .+4             ;BRANCH IF OK
 2342  010324  104001                  HLT+1                   ;WRONG LINE # RECEIVED
 2343                                                          ;R1 = CONTENTS OF RBUF
 2344                                                          ;BITS8-11 = LINE #
 2345  010326  117702  170754  5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
 2346  010332  130201                  BITB    R2,     R1      ;CHECK CHAR LENGTH.
 2347  010334  001401                  BEQ     .+4             ;BRANCH IF OK
 2348  010336  104002                  HLT+2                   ;WRONG CHARACTER LENGTH
 2349                                                          ;R1=DATA FROM FI/FO
 2350                                                          ;R2=MASK (BITS SET NOT EXPECTED)
 2351  010340  105102  6$:     COMB    R2                      ;REVERSE THE MASK
 2352  010342  120102                  CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
 2353  010344  001401                  BEQ     .+4             ;BRANCH IF OK
 2354  010346  104002                  HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
 2355                                                          ;R1=DATA FROM FI/FO (COMPLETE WORD)
 2356                                                          ;R2=DATA (LOW BYTE) EXPECTED
 2357  010350  017701  170636  7$:     MOV     @RBLF,  R1      ;READ FI/FO
 2358  010354  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
 2359  010356  104001                  HLT+1                   ;CHARACTER PRESENT STAYED SET
 2360                                                          ;R1 = CONTENTS OF RBUF
 2361  010360  017701  170624  8$:     MOV     @CSR,   R1      ;SAVE THE CSR
 2362  010364  022701  100405          CMP     #100405,R1      ;CHECK THE CSR
 2363  010370  001401                  BEQ     .+4             ;BRANCH IF OK
 2364  010372  104001                  HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
 2365                                                          ;R1 = CONTENTS OF CSR
 2366  010374  005077  170614          CLR     @TCR            ;CLEAR TCR
 2367  010400  005077  170604          CLR     @CSR            ;CLEAR CSR
 2368  010404  104400                  SCOPE
 2369
 2370                          ;*****************************************************************
 2371                          ;TEST 45:       TEST THAT LINE 11 CAN TRANSMIT AND
 2372                          ;               RECEIVE A CHARACTER.
 2373                          ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
 2374                          ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
 2375                          ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
 2376                          ;       4$:  CHECKS THAT RIGHT LINE # (11) IN FI/FO
 2377                          ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
 2378                          ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
 2379                          ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
 2380                          ;       8$:  CHECKS THAT DONE CLEARS
 2381                          ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DJ3 SERIES
 2382                          ;*****************************************************************
 2383
 2384                          ;               INITIALIZE
 2385                          ;       DEVICE"CSR"REGISTER
 2386                                                  SET:
 2387  010406  004737  015566  TST45:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
 2388                                                          ;BIT3 = CLEAR MOS
 2389                                                          ;BIT8 = MASTER XMTR SCAN ENB
 2390                          ;WAIT FOR MOS TO CLEAR
 2391                                                  SET:
 2392                          ;                               ;BIT0 = RECEIVER ENABLE
 2393                          ;
```

```
2394  010412  352777  004000  170574          BIS     #BIT11, @TCR    ;SET XMTR CONTROL BIT, LINE 11
2395  010420  017701  170564          LOP45:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
2396  010424  100375                          BPL     LOP45
2397  010426  012777  000377  170552          MOV     #377.   @TBUF   ;SEND A RUBOUT
2398  010434  005000                          CLR     R0              ;CLEAR COUNTER
2399  010436  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
2400  010440  001376                          BNE     1$
2401  010442  105777  170542                  TSTB    @CSR            ;WAIT FOR DONE
2402  010446  100405                          BMI     2$              ;BRANCH WHEN DONE
2403  010450  005200                          INC     R0              ;TIME COUNTER
2404  010452  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
2405  010454  017701  170530                  MOV     @CSR,   R1      ;SAVE CSR
2406  010460  104001                          HLT+1                   ;DONE NEVER CAME UP
2407                                                                  ;R1 = CONTENTS OF CSR
2408  010462  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
2409  010466  017701  170520                  MOV     @RBUF,  R1      ;READ THE FI/FO
2410  010472  100401                          BMI     .+4             ;BRANCH IF CHARACTER READY
2411  010474  104001                          HLT+1                   ;CHARACTER READY DIDN'T SET
2412                                                                  ;R1 = CONTENTS OF RBUF
2413  010476  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
2414  010502  001401                          BEQ     .+4             ;BRANCH IF NONE
2415  010504  104001                          HLT+1                   ;ERROR IN RECEIVED CHAR
2416                                                                  ;R1 = CONTENTS OF RBUF
2417                                                                  ;BIT14=UART OVERRUN
2418                                                                  ;BIT13=FRAMING ERROR
2419                                                                  ;BIT12=PARITY ERROR
2420  010506  010102                  4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
2421  010510  042702  170377                  BIC     #170377,R2      ;MASK LINE#
2422  010514  000302                          SWAB    R2              ;LINE # IN LOW BYTE
2423  010516  122702  000013                  CMPB    #11.,   R2      ;CHECK LINE #
2424  010522  001401                          BEQ     .+4             ;BRANCH IF OK
2425  010524  104001                          HLT+1                   ;WRONG LINE # RECEIVED
2426                                                                  ;R1 = CONTENTS OF RBUF
2427                                                                  ;BITS8-11 = LINE #
2428  010526  117702  170554          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
2429  010532  130201                          BITB    R2,     R1      ;CHECK CHAR LENGTH.
2430  010534  001401                          BEQ     .+4             ;BRANCH IF OK
2431  010536  104002                          HLT+2                   ;WRONG CHARACTER LENGTH
2432                                                                  ;R1=DATA FROM FI/FO
2433                                                                  ;R2=MASK (BITS SET NOT EXPECTED)
2434  010540  105102                  6$:     COMB    R2              ;REVERSE THE MASK
2435  010542  120102                          CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
2436  010544  001401                          BEQ     .+4             ;BRANCH IF OK
2437  010546  104002                          HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
2438                                                                  ;R1=DATA FROM FI/FO (COMPLETE WORD)
2439                                                                  ;R2=DATA (LOW BYTE) EXPECTED
2440  010550  017701  170436          7$:     MOV     @RBUF,  R1      ;READ FI/FO
2441  010554  100001                          BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
2442  010556  104001                          HLT+1                   ;CHARACTER PRESENT STAYED SET
2443                                                                  ;R1 = CONTENTS OF RBUF
2444  010560  017701  170424          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
2445  010564  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
2446  010570  001401                          BEQ     .+4             ;BRANCH IF OK
2447  010572  104001                          HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
2448                                                                  ;R1 = CONTENTS OF CSR
2449  010574  005 77  170414                  CLR     @TCR            ;CLEAR TCR
```

```
                                                CLR     @CSR            ;CLEAR CSR
                                                SCOPE
                                                INC     DJLEN
                                                MOV     #.+6.   LAD

                                        ;*****************************************************************
                                        ;TEST 46:       TEST THAT LINE 12 CAN TRANSMIT AND
                                        ;               RECEIVE A CHARACTER. (377)
                                        ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                                        ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
                                        ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
                                        ;       4$:  CHECKS THAT RIGHT LINE # (12) IN FI/FO
                                        ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                                        ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                                        ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                                        ;       8$:  CHECKS THAT DONE CLEARS
                                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DC3 SERIES
                                        ;*****************************************************************

                                        ;               INITIALIZE
                                        ;       DEVICE"CSR"REGISTER
                                                                        SET:
010620  004737  015566          ST46:   JSR     PC,     @#INITD         ;BIT2 = MAINTENANCE
                                                                        ;BIT3 = CLEAR MOS
                                                                        ;BIT8 = MASTER XMTR SCAN ENB
                                        ;WAIT FOR MOS TO CLEAR
                                        ;
                                                                        SET:
                                        ;                               ;BIT0 = RECEIVER ENABLE
                                        ;
010624  052777  010000  170362          BIS     #BIT12, @TCR    ;SET XMTR CONTROL BIT, LINE 12
010632  017701  170352          LOP46:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
010636  100375                  BPL     LOP46
010640  012777  000377  170350  MOV     #377,   @TBUF   ;SEND A RUBOUT
010646  005000                  CLR     R0              ;CLEAR COUNTER
010650  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
010652  001376                  BNE     1$
010654  105777  170330          TSTB    @CSR            ;WAIT FOR DONE
010660  100405                  BMI     2$              ;BRANCH WHEN DONE
010662  005200                  INC     R0              ;TIME COUNTER
010664  001371                  BNE     1$              ;BRANCH IF NOT TIME-OUT
010666  017701  170316          MOV     @CSR,   R1      ;SAVE CSR
010672  104001                  HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
010674  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
010700  017701  170306          MOV     @RBUF,  R1      ;READ THE FI/FO
010704  100401                  BMI     .+4             ;BRANCH IF CHARACTER READY
010706  104001                  HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
010710  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
010714  001401                  BEQ     .+4             ;BRANCH IF NONE
010716  104001                  HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR
010720  010102          4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
```

```
                        ;TEST 46:  TEST ALL OF LINE 12 TRANSMIT AND RECEIVE LOGIC

                                   MOV     #170377,R2      ;MASK LINE#
                                   MOVB    R2,R2           ;LINE # IN LOW BYTE
                                   CMPB    #12..,  R2      ;CHECK LINE #
                                   BEQ     .+4             ;BRANCH IF OK
                                   HLT     +1              ;WRONG LINE # RECEIVED
                                                           ;R1 = CONTENTS OF RBUF
                                                           ;BITS9-11 = LINE #
                                   BIT     #BCCLEN, R0     ;GET MASK OF CHARACTER
                                   BNE     .+4             ;CHECK CHAR LENGTH.
                                   HLT     +0              ;BRANCH IF OK
                                                           ;WRONG CHARACTER LENGTH
                                                           ;R1=DATA FROM FIFO
                                                           ;R2=MASK BITS SET NOT EXPECTED
                                   COM     R2              ;REVERSE THE MASK
                                   BIT     R1,     R2      ;CHECK THE ACTURAL DATA
                                   BEQ     .+4             ;BRANCH IF OK
                                   HLT     +0              ;WRONG CHAR LEN OR DATA ERROR
                                                           ;R1=DATA FROM FIFO (COMPLETE WORD
                                                           ;R2=DATA (LOW BYTE) EXPECTED
                                   BIT     #RBUF,  R1      ;READ FIFO
                                   BNE     .+4             ;BRANCH IF CHAR PRESENT NOT SET
                                   HLT     +1              ;CHARACTER PRESENT STATUS BIT
                                                           ;R1 = CONTENTS OF RBUF
                                   MOV     #XCSR,  R2      ;SAVE THE CSR
                                   CMP     #CSRINI, R2     ;CHECK THE CSR
                                   BEQ     .+4             ;BRANCH IF OK.
                                   HLT     +1              ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                           ;R2 = CONTENTS OF CSR
                                   CLR     @TCR            ;CLEAR TCR
                                   CLR     @XCSR           ;CLEAR CSR
                                   SCOPE

                        ;*****************************************************************
                        ;TEST 47:  TEST THAT LINE 12 CAN TRANSMIT AND
                        ;          RECEIVE A CHARACTER. (INT)
                        ;
                        ;          CHECKS THAT DONE SETS IN REASONABLE TIME.
                        ;          CHECKS THAT CHAR PRESENT SET IN FIFO
                        ;          CHECKS THAT NO ERRORS IN FIFO
                        ;          CHECKS THAT RIGHT LINE # (12) IN FIFO
                        ;          CHECKS FOR RIGHT CHARACTER LENGTH
                        ;          CHECKS THAT CORRECT DATA WAS RECEIVED
                        ;          CHECKS THAT CHARACTER PRESENT CLEARS
                        ;          CHECKS THAT DONE CLEARS
                        ;PROBABLE FAULTY LOGIC:  M7285 (12-)  ALL; M7279 ALL; UART CARD DD3 SERIES
                        ;*****************************************************************

                                   ;INITIALIZE
                                   ;       DEVICE CSR REGISTER
                                                           ;       SET:
C11020  004737  C15566  T547:      JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                           ;BIT3 = CLEAR MOS
                                                           ;BIT8 = MASTER SCAN ENB
                                   ;WAIT FOR MOS TO CLEAR
                                                           ;       SET:
                                                           ;BIT0 = RECEIVER ENABLE
```

```
                        ;TEST ALL OF LINE 13 TRANSMIT AND RECEIVE LOGIC

                        ;SET XMTR CONTROL BIT, LINE 13
                        ;WAIT FOR XMTR READY
                        ;SEND A RUBOUT
                        ;CLEAR COUNTER
                        ;SHORT WAIT LOOP
                        ;WAIT FOR DONE
                        ;BRANCH WHEN DONE
                        ;TIME COUNTER
                        ;BRANCH IF NOT TIME-OUT
                        ;SAVE CSR
                        ;DONE NEVER CAME UP
                        ;R1 = CONTENTS OF CSR
                        ;SAVE TIMER
                        ;READ THE FIFO
                        ;BRANCH IF CHARACTER READY
                        ;CHARACTER READY DIDN'T SET
                        ;R1 = CONTENTS OF RBUF
                        ;CHECK FOR ERROR BITS
                        ;BRANCH IF NONE
                        ;ERROR IN RECEIVED CHAR
                        ;R1 = CONTENTS OF RBUF
                        ;BIT14=UART OVERRUN
                        ;BIT13=FRAMING ERROR
                        ;BIT12=PARITY ERROR
                        ;DUPLICATE DATA WORD
                        ;MASK LINE#
                        ;LINE # IN LOW BYTE
                        ;CHECK LINE #
                        ;BRANCH IF OK
                        ;WRONG LINE # RECEIVED
                        ;R1 = CONTENTS OF RBUF
                        ;BITS8-11 = LINE #
                        ;GET MASK OF CHARACTER
                        ;CHECK CHAR LENGTH.
                        ;BRANCH IF OK
                        ;WRONG CHARACTER LENGTH
                        ;R1=DATA FROM FIFO
                        ;R2=MASK OF BITS WE'RE NOT EXPECTING
                        ;REVERSE THE MASK
                        ;CHECK THE ACTURAL DATA
                        ;BRANCH IF OK
                        ;WRONG CHAR LEN OR DATA ERROR
                        ;R1=DATA FROM FIFO INCLUDING ALL BITS
                        ;R2=DATA (LOW BYTE) EXPECTED
                        ;READ FIFO
                        ;BRANCH IF CHAR PRESENT NOT SET
                        ;CHARACTER PRESENT ERROR BIT SET
                        ;R1 = CONTENTS OF RBUF
                        ;SAVE THE CSR
                        ;CHECK THE CSR
                        ;BRANCH IF OK
                        ;THE [illegible] ERROR OR OTHER CSR ERROR
                        ;R1 = CONTENTS OF CSR
                        ;[illegible]
```

```
011616  000377  167772          CLR     @CSR            ;CLEAR CSR
011622  104400                  SCOPE

                        ;*****************************************************************
                        ;TEST 52:       TEST THAT LINE 14 CAN TRANSMIT AND
                        ;               RECEIVE A CHARACTER. (377)
                        ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                        ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
                        ;       3$:  CHECKS THAT NO ERRORS IN FI FO
                        ;       4$:  CHECKS THAT RIGHT LINE # (14) IN FI FO
                        ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
                        ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
                        ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
                        ;       8$:  CHECKS THAT DONE CLEARS
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DJ3 SERIES
                        ;*****************************************************************

                        ;               INITIALIZE
                        ;       DEVICE"CSR"REGISTER
                                                        SET:
011632  004737  011666  5$:     JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MCS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
                        ;WAIT FOR MCS TO CLEAR
                        ;                               SET:
                        ;                               ;BIT0 = RECEIVER ENABLE
                        ;
011636  050477  040000  167752          BIS     #BIT14, @TCR    ;SET XMTR CONTROL BIT, LINE 14
011642  105777  167752  10$:    TSTB    @CSR,   R1      ;WAIT FOR XMTR READY
011646  100375                  BPL     10$
011650  012777  000377  167750          MOV     #377,   @TBUF   ;SEND A RUBOUT
011656  005002                  CLR     R2              ;CLEAR COUNTER
011660  005202                  INC     R2      15$:    ;SHORT WAIT LOOP
011662  001376                  BNE     15$
011664  105777  167730          TSTB    @CSR            ;WAIT FOR DONE
011670  100411                  BMI     20$             ;BRANCH WHEN DONE
011672  005202                  INC     R2              ;TIME COUNTER
011674  001373                  BNE     15$             ;BRANCH IF NOT TIME-OUT
011676  013701  167716          MOV     @CSR,   R1      ;SAVE CSR
011702  104001                  HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
011704  050003                  BIS     R0,     TIMER   ;SAVE TIMER
011706  013701  167726  20$:    MOV     @RBUF,  R1      ;READ THE FIFO
011712  100411                  BMI     .+4             ;BRANCH IF CHARACTER READY
011714  104001                  HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
011716  032701  070000          BIT     #70000, R1      ;CHECK FOR ERROR BITS
011722  001401                  BEQ     .+4             ;BRANCH IF NONE
011724  104001                  HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR
011726  010102                  MOV     R1,     R2      ;DUPLICATE DATA WORD
011730  042702  170377          BIC     #170377,R2      ;MASK LINE#
011734  000302                  SWAB    R2              ;LINE # IN LOW BYTE
```

```
011330  012703  000016          CMPB    #14.,   R2      ;CHECK LINE #
011334  001401                  BEQ     .+4             ;BRANCH IF OK
011336  104001                  HLT+1                   ;WRONG LINE # RECEIVED
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 = LINE #

011340  011702  167742  55:     MOVB    DZJLEN, R2      ;GET MASK OF CHARACTER
011344  130201                  BITB    R2,     R1      ;CHECK CHAR LENGTH.
011346  001401                  BEQ     .+4             ;BRANCH IF OK
011350  104002                  HLT+2                   ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FIFO
                                                        ;R2=MASK BITS SET NOT EXPECTED
                                                        ;REVERSE THE MASK
                                  65:     COMB    R2
                                          CMPB    R1,     R2      ;CHECK THE RETURNED DATA
                                          BEQ     .+4             ;BRANCH IF OK
                                          HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                                  ;R1=DATA FROM FIFO INCOMPLETE WORD
                                                                  ;R2=DATA LOW BYTE EXPECTED
                                                                  ;READ FIFO
                                  75:     MOV     @RBUF,  R1      ;BRANCH IF CHAR PRESENT NOT SET
                                          BPC     .+4             ;CHARACTER PRESENT FLAGGED SET
                                          HLT+1                   ;R1 = CONTENTS OF RBUF
                                                                  ;SAVE THE CSR
                                  85:     MOV     @CSR,   R1      ;CHECK THE CSR
                                          CMP     #100405,R1      ;BRANCH IF OK
                                          BEQ     .+4             ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                          HLT+1                   ;R1 = CONTENTS OF CSR
                                                                  ;CLEAR TCR
                                          CLR     @TCR            ;CLEAR CSR
                                          CLR     @CSR
                                          SCOPE
```

```
                        ;*****************************************************************
                        ;TEST 51:       TEST THAT LINE 15 CAN TRANSMIT AND
                        ;               RECEIVE A CHARACTER. (377)
                        ;       15:  CHECKS THAT DONE SETS IN REASONABLE TIME.
                        ;       25:  CHECKS THAT CHAR PRESENT IS IN FIFO
                        ;       35:  CHECKS THAT NO ERRORS IN FIFO
                        ;       45:  CHECKS THAT RIGHT LINE # (15) IN FIFO
                        ;       55:  CHECKS FOR RIGHT CHARACTER LENGTH
                        ;       65:  CHECKS THAT CORRECT DATA WAS RECEIVED
                        ;       75:  CHECKS THAT CHARACTER PRESENT CLEARS
                        ;       85:  CHECKS THAT DONE CLEARS
                        ;PROBABLE FAULTY LOGIC:  M7285 (DZ-11) ALL; M7279 ALL; QUART CARD DCJ SERIES
                        ;*****************************************************************

                        ;               INITIALIZE
                        ;       DEVICE CSR REGISTER
                                                        SET:
011420  004737  015566  5$:     JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
                        ;WAIT FOR MOS TO CLEAR
                                                        SET:
                        ;                               ;BITC = RECEIVER ENABLE
                        ;
011424  052777  100008  167562          BIS     #BIT15, @TCR    ;SET XMTR CONTROL BIT, LINE 15
011430  017701  167562  10$:    MOV     @CSR,   R1      ;WAIT FOR XMTR READY
```

```
011436  100375          BPL     .-6
011440  012737  000377  167550  MOV     #377.,  @TBUF   ;SEND A RUBOUT
011446  005000          CLR     R0                      ;CLEAR COUNTER
011450  005305          DEC     R5              1$:     ;SHORT WAIT LOOP
011452  001375          BNE     .-4
011454  105737  167530  TSTB    @CSR                    ;WAIT FOR DONE
011460  100405          BMI     3$                      ;BRANCH WHEN DONE
011462  005200          INC     R0                      ;TIME COUNTER
011464  001371          BNE     1$                      ;BRANCH IF NOT TIME-OUT
011466  017701  167516  MOV     @CSR,   R1              ;SAVE CSR
011472  104001          HLT+1                           ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
011474  050037  001302  BIS     R0,     TIMER   2$:     ;SAVE TIMER
011500  017701  167506  MOV     @RBUF,  R1              ;READ THE FIFO
011504  100401          BMI     .+4                     ;BRANCH IF CHARACTER READY
011506  104001          HLT+1                           ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
011510  032701  070000  BIT     #70000, R1      3$:     ;CHECK FOR ERROR BITS
011514  001401          BEQ     .+4                     ;BRANCH IF NONE
011516  104001          HLT+1                           ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR
011520  010102          MOV     R1,     R2      4$:     ;DUPLICATE DATA WORD
011522  042702  170377  BIC     #170377,R2              ;MASK LINE#
011526  000302          SWAB    R2                      ;LINE # IN LOW BYTE
011530  122702  000017  CMPB    #15.,   R2              ;CHECK LINE #
011534  001401          BEQ     .+4                     ;BRANCH IF OK
011536  104001          HLT+1                           ;WRONG LINE # RECEIVED
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 = LINE #
011540  117202  167542  MOVB    @DDJLEN, R2     5$:     ;GET MASK OF CHARACTER
011544  130201          BITB    R2,     R1              ;CHECK CHAR LENGTH.
011546  001401          BEQ     .+4                     ;BRANCH IF OK
011550  104002          HLT+2                           ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FIFO
                                                        ;R2=MASK (BITS SET NOT EXPECTED)
011552  105102          COMB    R2              6$:     ;REVERSE THE MASK
011554  120102          CMPB    R1,     R2              ;CHECK THE ACTURAL DATA
011556  001401          BEQ     .+4                     ;BRANCH IF OK
011560  104002          HLT+2                           ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FIFO (COMPLETE WORD)
                                                        ;R2=DATA (LOW BYTE) EXPECTED
011562  017701  167424  MOV     @RBUF,  R1      7$:     ;READ FIFO
011566  100001          BPL     .+4                     ;BRANCH IF CHAR PRESENT NOT SET
011570  104001          HLT+1                           ;CHARACTER PRESENT STATED SET
                                                        ;R1 = CONTENTS OF RBUF
011572  017701  167412  MOV     @CSR,   R1      8$:     ;SAVE THE CSR
011576  022701  100405  CMP     #100405,R1              ;CHECK THE CSR
011602  001401          BEQ     .+4                     ;BRANCH IF OK
011604  104001          HLT+1                           ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                        ;R1 = CONTENTS OF CSR
011606  005077  167402  CLR     @TCR                    ;CLEAR TCR
011612  005077  167372  CLR     @CSR                    ;CLEAR CSR
011616  104400          SCOPE
```

```
011620  162737  000003  001306          SUB     #3,     DJLEN
011626  005237  001302                  INC     TIMER           ;WORSE CASE TIME, ONE CHARACTER
011632  013700  001300                  MOV     TIMER,  R0      ;DUP TIMER
011636  006200                          ASR     R0              ;/2
011640  006200                          ASR     R0              ; 4
011642  005200                          INC     R0              ;+1
011644  006137  001302                  ROL     TIMER           ;TIMER * 2
011650  060037  001302                  ADD     R0,     TIMER   ;2.25 TIMES ONE CHARACTER TIME
011654  012737  011662  015770          MOV     #.+6,   LAD     ;RESET LOOP ADDRESS

                                ;**********************************************************************
                                ;TEST 52:       TEST THAT CHARACTER PRESENT (BIT15) OF RBUF
                                ;               IS CLEARED (BY CLEAR MOS).
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-8: E17,E14,E15; M7279; M7280
                                ;**********************************************************************

                                ;       INITIALIZE
                                ;
011660  004737  015566          ;ST52:  JSR     PC,@#INITD
                                ;SET:
                                ;       BIT2 = MAINTENANCE
                                ;       BIT3 = CLEAR MOS
                                ;       BIT8 = TRANS SCAN ENABLE
                                ;       WAIT FOR BIT4 = MOS CLEAR
                                ;SET:
                                ;       BIT0 = RECEIVER ENABLE
011666  012777  000005  167320          MOV     #BIT0,  @TCR    ;TRANS CONTROL, LINE0
011674  012704  000004                  MOV     #4,R4           ;SET UP COUNTER
011700  105711  167314          1$:     MOVB    @TCSR,R1        ;WAIT FOR TRANS READY
011704  100375                          BPL     1$
011706  116467  167304  001300          MOVB    R4,.TBUF        ;TRANSMIT COUNT
011714  005304                          DEC     R4              ;COUNT DOWN
011716  001371                          BNE     1$

011720  105777  167266          2$:     TSTB    @TCSR           ;MAKE SURE DONE IS SET
011724  100375                          BPL     2$

011726  052777  000010  167260          BIS     #BIT3,@TCSR     ;CLEAR MOS
011734  032777  000020  167252  3$:     BIT     #BIT4,@TCSR     ;WAIT FOR CLRMOS TO FINISH
011742  001374                          BNE     3$

011744  017701  167244                  MOV     @RBUF,R1        ;CHECK RBUF AND SAVE
011750  100001                          BPL     .+4
011752  000001                          WAIT
011754  010102                          MOV     R1,R2
011756  017700  167230                  MOV     @TCSR,R0        ;SAVE CSR
011762  042700  100400                  BIC     #100400,R0      ;CHECK CSR
011766  001401                          BEQ     .+4             ;BRANCH IF OK
011770  000001                          WAIT                    ;CSR ERROR, POSSIBILITIES:
                                                                ;(1) DONE DIDN'T CLEAR

                                                                ;(2) TRANSMITTER READY DIDN'T CLR.
                                                                ;R1=CONTENTS OF CSR

011772  012737  000302                  MOV     #TIMER, PC      ;SET UP TIMER
012000  005505                  4$:     DECB    R5              ;SHORT WAIT LOOP
012002  001376                          BNE     4$
```

```
012206  167206          MOV     @CSR,R1         ;SAVE CSR
012212  105701          TSTB    R1              ;CHECK FOR DONE
012214  100404          BMI     .+4             ;BRANCH IF OK
012216  104401          HLT+1                   ;DONE CAME UP!
                                                ;MOS MUST NOT HAVE CLEARED
012220  005300          DEC     R0              ;TIMER COUNT
012222  001367          BNE     4$

012224  022701  100405  CMP     #100405,R1      ;CHECK CSR
012230  001401          BEQ     .+4             ;BRANCH IF OK
012232  104001          HLT+1                   ;CSR ERROR

012234  017701  167162  MOV     @RBUF,R1        ;CHECK RBUF
012240  100001          BPL     .+4             ;BRANCH IF OK
012242  104001          HLT+1                   ;RBUF NOT EMPTY!

012244  005077  167154  CLR     @TCR
012250  005077  167144  CLR     @CSR

012254  104400          SCOPE


                        ;*****************************************************************
                        ;TEST 53:       TEST THAT TRANSMITTER READY CLEARS WHEN TBUF IS LOADED
                        ;       NOTE:   DUE TO THE DOUBLE BUFFERING BY THE UART, TBUF IS LOADED TWO
                        ;                 MUST BE LOADED TO INSURE SEEING TRANSMITTER READY CLEAR
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-6) E39, E23, E49
                        ;*****************************************************************

                        ;               INITIALIZE
                        ;       DEVICE CSR REGISTER
                        ;                                               SET:
012256  004737  015566  TST53:  JSR     PC,     @#INITD                 ;BIT2 = MAINTENANCE
                                                                        ;BIT3 = CLEAR MOS
                                                                        ;BIT5 = MASTER XMTR SCAN ENB
                        ;WAIT FOR MOS TO CLEAR
                        ;                                               SET:
                        ;                                               ;BIT0 = RECEIVER ENABLE
                        ;
012262  052777  000001  167134          BIS     #BIT0,  @TCR    ;TRANS CONTROL, LINE 0
012270  017701  167124  1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
012274  100375          BPL     1$
012276  012777  000001  167122          MOV     #1,     @TBUF   ;TRANSMIT A 1
012304  017701  167110  2$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
012310  100375          BPL     2$
012312  012777  000002  167106          MOV     #2,     @TBUF   ;TRANSMIT A 2
012320  017701  167074          MOV     @CSR,   R1      ;CHECK FOR XMTR READY
012324  100001          BPL     .+4             ;BRANCH IF XMTR READY CLEARED
012326  104001          HLT+1                   ;TRANSMITTER READY FAILED TO CLEAR
                                                ;R1 = CONTENTS OF CSR
012330  013700  001302          MOV     TIMER,  R0      ;SET UP TIMER
012334  105305          3$:     DECB    R5              ;SHORT WAIT LOOP
012336  001376          BNE     3$
012340  017701  167054          MOV     @CSR,   R1      ;SAVE CSR FOR THE RECORD
012344  000400          BR      .+2             ;NOP FOR TIMING
```

```
002898  012136  005300          DEC     R0              ;TIMER COUNT
002899  012140  001371          BNE     3$              ;BRANCH IF MORE TIME
002900
002901  012142  022701  100605  CMP     #100605,R1      ;CHECK CSR
002902  012146  001401          BEQ     .+4             ;BRANCH IF OK
002903  012150  104001          HLT+1                   ;DONE DIDN'T SET OR OTHER CSR ERROR
002904                                                  ;R1 = CONTENTS OF CSR
002905  012152  017701  167034  MOV     @RBUF,  R1      ;CHECK RBUF FOR CHAR PRESENT
002906  012156  100401          BMI     .+4             ;BRANCH IF CHAR PRESENT
002907  012160  104001          HLT+1                   ;CHAR PRESENT MISSING
002908                                                  ;R1 = CONTENTS OF RBUF
002909  012162  022701  100001  CMP     #100001,R1      ;CHECK THE DATA
002910  012166  001401          BEQ     .+4             ;BRANCH IF OK
002911  012170  104001          HLT+1                   ;RECEIVER ERROR
002912                                                  ;R1 = CNTENTS OF RBUF
002913  012172  017701  167014  MOV     @RBUF,  R1      ;CHECK RBUF FOR SECOND CHAR
002914  012176  100401          BMI     .+4             ;BRANCH IF CHAR PRESENT
002915  012200  104001          HLT+1                   ;CHAR PRESENT MISSING
002916                                                  ;R1 = CONTENTS OF RBUF
002917  012202  022701  100002  CMP     #100002,R1      ;CHECK THE DATA
002918  012206  001401          BEQ     .+4             ;BRANCH IF OK
002919  012210  104001          HLT+1                   ;RECEIVER ERROR
002920                                                  ;R1 = CNTENTS OF RBUF
002921  012212  017701  166774  MOV     @RBUF,  R1      ;CHECK FOR NO MORE CHARACTERS
002922  012216  100001          BPL     .+4             ;BRANCH IF CHAR PRESENT CLEARED
002923  012220  104001          HLT+1                   ;CHAR PRESENT NOT CLEAR!
002924                                                  ;R1 = CONTENTS OF RBUF
002925  012222  017701  166762  MOV     @CSR,   R1      ;SAVE CSR
002926  012226  022701  100405  CMP     #100405,R1      ;CHECK CSR
002927  012232  001401          BEQ     .+4             ;BRANCH IF OK
002928  012234  104001          HLT+1                   ;CSR ERROR
002929                                                  ;R1 = CONTENTS OF CSR
002930  012236  005077  166752  CLR     @TCR            ;CLEAR TCR
002931  012242  005077  166742  CLR     @CSR            ;CLEAR CSR
002932  012246  104400          SCOPE
002933
002934
002935                          ;*****************************************************************
002936                          ;TEST 54:       TEST THAT RECEIVER ENABLE ON A 0 INHIBITS DONE
002937                          ;                 AND CHARACTER PRESENT.
002938                          ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) E32, E15
002939                          ;*****************************************************************
002940
002941  012250  012777  000414  166732  TST54:  MOV     #414,   @CSR    ;BIT2 = MAINTENANCE
002942                                                                  ;BIT3 = CLEAR MOS
002943                                                                  ;BIT8 = MASTER TRAN SCAN ENB
002944  012256  032777  000020  166724  10$:     BIT     #BIT4,  @CSR    ;WAIT FOR MOS TO CLEAR
002945  012264  001374          BNE     10$
002946  012266  052777  000000  166720          BIS     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE0
002947  012274  017701  166710          1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
002948  012300  100375          BPL     1$
002949  012302  012777  000252  166706          MOV     #252,   @TBUF   ;SEND AN "*"
002950  012310  013700  001302          MOV     TIMER,  R0              ;SET UP TIMER
002951  012314  105305          2$:     DECB    R5                      ;SHORT WAIT LOOP
002952  012316  001376          BNE     2$
002953  012320  017701  166664          MOV     @CSR,   R1      ;SAVE CSR FOR TYPING
```

```
012324  105701                          TSTB    R1              ;CHECK FOR DONE
012326  100002                          BPL     3$              ;BRANCH IF NOT SET
012330  104001                          HLT+1                   ;DONE SET WHEN RCV ENB CLR
                                                                ;R1=CONTENTS OF CSR
012332  000402                          BR      4$
012334  005300                  3$:     DEC     R0              ;TIMER COUNT
012336  001366                          BNE     2$              ;BRANCH IF MORE TIMER

012340  017701  166646          4$:     MOV     @RBUF,  R1      ;CHECK AND SAVE FI/FO
012344  100001                          BPL     .+4             ;BRANCH IF OK
012346  104001                          HLT+1                   ;CHARACTER PRESENT IN FI FO
                                                                ;R1=DATA FROM FI/FO
012350  005277  166634                  INC     @CSR            ;SET RECEIVER ENABLE
012354  017701  166630          5$:     MOV     @CSR,   R1      ;SAVE CSR
012360  105701                          TSTB    R1              ;CHECK FOR DONE
012362  100403                          BMI     6$              ;BRANCH IF OK
012364  105300                          DECB    R0              ;SHORT TIMER
012366  001372                          BNE     5$
012370  104001                          HLT+1                   ;DONE DIDN'T COME UP WHEN RECEIVER ENABLED
                                                                ;UART SHOULD HAVE HELD A CHARACTER
                                                                ;R1 = CONTENTS OF CSR
012372  017701  166614          6$:     MOV     @RBUF,  R1      ;CHECK FOR CHARACTER PRESENT
012376  100401                          BMI     .+4             ;BRANCH IF OK
012400  104001                          HLT+1                   ;CHARACTER PRESENT MISSING
                                                                ;R1 = CONTENTS OF RBUF
012402  005077  166606                  CLR     @TCR            ;CLR TRANS CONTROL REG

012406  104400                          SCOPE

                        ;*****************************************************************
                        ;TEST 55:       TEST THAT HALF DUPLEX (BIT1) DISABLES THE RECEIVER UARTS.
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-4) E32, E17, E22, (D2-2) E5, E1
                        ;*****************************************************************

012410  004737  015556          TST55:  JSR     PC,     @#INITC ;INITIALIZE
                                                                ;BIT1 = HALF DUPLEX
                                                                ;BIT2 = MAINTENANCE
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT8 = MASTER TRAN SCAN ENB
                                                                ;WAIT FOR MOS TO CLEAR
                                                                ;BIT0 = RECEIVER ENABLE
012414  012777  000001  166572          MOV     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE0
012422  017701  166562          1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
012426  100375                          BPL     1$
012430  012777  000252  166560          MOV     #252,   @TBUF   ;SEND AN "*"
012436  013700  001302                  MOV     TIMER,  R0      ;SET UP TIMER
012442  105305                  2$:     DECB    R5              ;SHORT WAIT LOOP
012444  001376                          BNE     2$
012446  017701  166536                  MOV     @CSR,   R1      ;SAVE CSR
012452  105701                          TSTB    R1              ;CHECK FOR DONE
012454  100002                          BPL     3$              ;BRANCH IF NOT SET
012456  104001                          HLT+1                   ;DONE SET WHEN HALF DUPLEX (BIT1) SET
                                                                ;R1=CONTENTS OF CSR
012460  000402                          BR      4$
012462  005300                  3$:     DEC     R0              ;TIMER COUNT
012464  001366                          BNE     2$              ;BRANCH IF MORE TIMER
```

```
3010
3011  012466  017701  166520          4$:   MOV     @RBUF,  R1      ;CHECK AND SAVE FI/FO
3012  012472  100001                        BP      .+4             ;BRANCH IF OK
3013  012474  104001                        HLT+1                   ;CHARACTER PRESENT IN FI/FO
3014                                                                  ;R1=DATA FROM FI/FO
3015  012476  042777  000002  166504        BIC     #BIT1,  @CSR    ;CLEAR HALF DUPLEX BIT
3016  012504  000240                        NOP
3017  012506  000240                        NOP
3018  012510  000240                        NOP
3019  012512  000240                        NOP
3020  012514  017701  166472          5$:   MOV     @RBUF,  R1      ;CHECK FOR CHAR PRESENT
3021  012520  100001                        BPL     .+4             ;BRANCH IF CHAR NOT PRESENT
3022  012522  104001                        HLT+1                   ;CHAR PRESENT AFTER H/D CLEARED
3023                                                                  ;R1 = CONTENTS OF RBUF
3024  012524  005077  166464                CLR     @TCR            ;CLR TRANS CONTROL REG
3025
3026  012530  104400                        SCOPE
3027
3028
3029                                ;*********************************************************************
3030                                ;TEST 56:       TEST THAT RECEIVER INTERRUPT DOES NOT OCCUR AT LEVEL 5
3031                                ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
3032                                ;*********************************************************************
3033
3034  012532  012777  012620  166460  TST56: MOV     #ISR56, @RCVVEC ;SET UP XMTR INTERRUPT VECTOR
3035  012540  012777  000340  166454        MOV     #340,   @RCVLVL ;AT LEVEL 7
3036  012546  042737  000340  177776        BIC     #340,   @#PS    ;CLEAR PS LEVEL
3037  012554  052737  000240  177776        BIS     #240,   @#PS    ;SET PS TO LEVEL 5
3038  012562  004737  015546                JSR     PC,     @#INITB ;SET:
3039                                                                  ;BIT2 = MAINTENANCE
3040                                                                  ;BIT3 = CLEAR MOS
3041                                                                  ;BIT6 = RECEIVER INTERUPT ENABLE
3042                                                                  ;BIT8 = MASTER TRANS SCAN ENABLE
3043                                                                  ;WAIT FOR MOS TO CLEAR
3044                                                                  ;BIT0 = RECEIVER ENABLE
3045  012566  012777  000001  166420        MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
3046  012574  017701  166410          1$:   MOV     @CSR,   R1      ;WAIT
3047  012600  100375                        BPL     1$
3048  012602  012777  000025  166406        MOV     #25,    @TBUF   ;SEND #25
3049  012610  105777  166374          2$:   TSTB    @CSR            ;WAIT FOR DONE
3050  012614  100375                        BPL     2$
3051  012616  000404                        BR      END56           ;OK, BRANCH IF NO INTERRUPT
3052
3053  012620  104000                ISR56:  HLT                     ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 5
3054  012622  012716  012630                MOV     #END56,(SP)     ;MOVE NEW RTI ADR ONTO STACK
3055  012626  000002                        RTI
3056
3057  012630  017701  166356          END56:  MOV     @RBUF,  R1      ;READ THE CHARACTER
3058  012634  100401                        BMI     .+4             ;BRANCH IF CHAR PRESENT
3059  012636  104001                        HLT+1                   ;CHAR PRESENT MISSING
3060                                                                  ;R1 = CONTENTS OF RBUF
3061  012640  022701  100025                CMP     #100025,R1      ;CHECK THE DATA
3062  012644  001401                        BEQ     .+4             ;BRANCH IF OK
3063  012646  104001                        HLT+1                   ;RECEIVED DATA ERROR
3064                                                                  ;R1 = CONTENTS OF RBUF
3065  012650  013777  001222  166342        MOV     RCVLVL,@RCVVEC
```

```
3066  012656  012777  000004  166336          MOV     #IOT,   @RCVLVL
3067  012664  005077  166324                  CLR     @TCR
3068  012670  005077  166314                  CLR     @CSR
3069  012674  042737  000340  177776          BIC     #340,@#PS
3070
3071  012702  104400                          SCOPE
3072
3073
3074                                  ;*************************************************************************
3075                                  ;TEST 57:       TEST THAT RECEIVER INTERRUPT OCCURS AT LEVEL 4
3076                                  ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
3077                                  ;*************************************************************************
3078
3079  012704  012777  012774  166306  TST57:  MOV     #ISR57, @RCVVEC ;SET UP XMTR INTERRUPT VECTOR
3080  012712  012777  000340  166302          MOV     #340,   @RCVLVL ;AT LEVEL 7
3081  012720  042737  000340  177776          BIC     #340,   @#PS    ;CLEAR PS LEVEL
3082  012726  052737  000200  177776          BIS     #200,   @#PS    ;SET PS TO LEVEL 4
3083  012734  004737  015546                  JSR     PC,     @#INITB ;SET:
3084                                                                  ;BIT2 = MAINTENANCE
3085                                                                  ;BIT3 = CLEAR MOS
3086                                                                  ;BIT6 = RECEIVER INTERUPT ENABLE
3087                                                                  ;BIT8 = MASTER TRANS SCAN ENABLE
3088                                                                  ;WAIT FOR MOS TO CLEAR
3089                                                                  ;BIT0 = RECEIVER ENABLE
3090  012740  012777  000001  166246          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
3091  012746  017701  166236          1$:     MOV     @CSR,   R1      ;WAIT
3092  012752  100375                          BPL     1$
3093  012754  012777  000025  166234          MOV     #25,    @TBUF   ;SEND #25
3094  012762  105777  166222          2$:     TSTB    @CSR            ;WAIT FOR DONE
3095  012766  100375                          BPL     2$
3096  012770  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 4
3097  012772  000403                          BR      END57           ;CONTINUE
3098
3099  012774  012716  013002          ISR57:  MOV     #END57,(SP)     ;MOVE NEW RTI ADR ONTO STACK
3100  013000  000002                          RTI
3101
3102  013002  017701  166204          END57:  MOV     @RBUF,  R1      ;READ THE CHARACTER
3103  013006  100401                          BMI     .+4             ;BRANCH IF CHAR PRESENT
3104  013010  104001                          HLT+1                   ;CHAR PRESENT MISSING
3105                                                                  ;R1 = CONTENTS OF RBUF
3106  013012  022701  100025                  CMP     #100025,R1      ;CHECK THE DATA
3107  013016  001401                          BEQ     .+4             ;BRANCH IF OK
3108  013020  104001                          HLT+1                   ;RECEIVED DATA ERROR
3109                                                                  ;R1 = CONTENTS OF RBUF
3110  013022  013777  001222  166170          MOV     RCVLVL,@RCVVEC
3111  013030  012777  000004  166164          MOV     #IOT,   @RCVLVL
3112  013036  005077  166152                  CLR     @TCR
3113  013042  005077  166142                  CLR     @CSR
3114  013046  042737  000340  177776          BIC     #340,@#PS
3115
3116  013054  104400                          SCOPE
3117
3118
3119                                  ;*************************************************************************
3120                                  ;TEST 60:       TEST FIFO OVERRUN
3121                                  ;               THE FIFO BUFFER SHOULD HOLD 64 CHARACTERS.
```

```
                                        ;PROBABLE FAULTY LOGIC:  M7285 (D1-7) E32, E17, E22 (D2-2) E5, E1
                                        ;*****************************************************************

                                        ;              INITIALIZE
                                        ;       DEVICE"CSR"REGISTER
                                        ;                                              SET:
013056  004737  015565          ST60:   JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT8 = MASTER XMTR SCAN ENB
                                        ;WAIT FOR MOS TO CLEAR
                                        ;                                              SET:
                                        ;                                              ;BIT0 = RECEIVER ENABLE
                                        ;
013062  012777  177777  166124          MOV     #177777,@TCR    ;TRANS CONTROL BIT, ALL LINES
013070  012700  000100                  MOV     #100.,  R0      ;SET UP COUNTER - 64. CHAR FI/FO BUFF
013074  017701  166110          1$:     MOV     @CSR,   R1      ;SAVE AND WAIT FOR TRANS READY
013100  100375                          BPL     1$
013102  000377  166110                  SWAB    @TBUF           ;TRANSMIT LINE # ON LINE
013106  032701  020000                  BIT     #BIT13, R1      ;CHECK FI/FO OVERRUN
013112  001401                          BEQ     .+4             ;BRANCH IF OK
013114  104001                          HLT+1                   ;FI/FO OVERRUN TOO SOON
                                                                ;R1=CONTENTS OF CSR
013116  005300                          DEC     R0              ;COUNT DOWN
013120  001365                          BNE     1$

013122  013700  001302                  MOV     TIMER,  R0      ;SET UP TIMER
013126  017701  166056          2$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
013132  100375                          BPL     2$
013134  105305                  3$:     DECB    R5              ;SHORT WAIT LOOP
013136  001376                          BNE     3$
013140  017701  166044                  MOV     @CSR,   R1      ;SAVE CSR FOR THE RECORD
013144  100401                          BMI     .+4             ;BRANCH IF TRANS READY
013146  104001                          HLT+1                   ;TRANS READY MISTERIOUSLY SET
                                                                ;R1=CONTENTS OF CSR
013150  005300                          DEC     R0              ;TIME 3 CHARACTER LENGTHS
013152  001370                          BNE     3$              ;BRANCH IF MORE TIME

                                        ;FI/FO SHOULD NOW BE FULL

013154  105701                          TSTB    R1              ;CHECK THAT DONE IS SET
013156  100401                          BMI     .+4             ;BRANCH IF OK
013160  104001                          HLT+1                   ;DONE DIDN'T COME UP!
                                                                ;R1 = CONTENTS OF CSR
013162  022701  100605                  CMP     #100605,R1      ;CHECK THAT FI/FO NOT OVERRUN
013166  001401                          BEQ     .+4             ;BRANCH IF OK
013170  104001                          HLT+1                   ;FI/FO OVERRUN SET
                                                                ;OR SOME OTHER CSR ERROR
                                                                ;R1=CONTENTS OF CSR

                                        ;*****************************************************************
                                        ;TEST 60A:      TEST THAT FI/FO OVERRUN COMES UP WHEN 65TH CHARACTER
                                        ;                 IS RECEIVED WITHOUT READING FI/FO
                                        ;*****************************************************************

013172  000377  166020          T60A:   SWAB    @TBUF           ;SEND 65TH CHARACTER
013176  013700  001302                  MOV     TIMER,  R0      ;SET UP TIMER
```

```
3178  013202  105305                  11$:    DECB    R5              ;SHORT WAIT LOOP
3179  013204  001376                          BNE     11$
3180  013206  017701  165776                  MOV     @CSR,R1         ;SAVE CSR
3181  013212  032701  020000                  BIT     #BIT13,R1       ;CHECK FI/FO OVERRUN
3182  013216  001003                          BNE     12$             ;BRANCH WHEN SET
3183  013220  005300                          DEC     R0              ;TIMER
3184  013222  001367                          BNE     11$             ;BRANCH IF MORE TIME
3185  013224  104001                          HLT+1                   ;FI/FO OVERRUN DIDN'T COMEUP
3186                                                                  ;R1 = CONTENTS OF CSR
3187  013226  022701  120605          12$:    CMP     #120605,R1      ;CHECK TOTAL CSR
3188  013232  001401                          BEQ     .+4             ;BRANCH IF OK
3189  013234  104001                          HLT+1                   ;SOMETHING IN CSR FOULED UP
3190                                                                  ;R1=CONTENTS OF CSR
3191
3192                          ;*****************************************************************
3193                          ;TEST 60B:      TEST THAT READING THE RECEIVER BUFFER CAUSES FI/FO
3194                          ;                 OVERRUN TO CLEAR.
3195                          ;       NOTE:   BECAUSE OF TIMING OF THE FI/FO, FI/FO OVERRUN CAN COME
3196                          ;                 BACK UP AFTER READING ONE CHARACTER, SO A SECOND MUST
3197                          ;                 BE READ TO INSURE THAT FI/FO OVERRUN IS CLEAR.
3198                          ;*****************************************************************
3199
3200  013236  012700  000002  T60B:   MOV     #2,     R0      ;SET UP COUNTER - 2 CHARACTERS
3201  013242  017701  165744  21$:    MOV     @RBUF,  R1      ;CHECK AND SAVE FIRST CHAR IN FI/FO
3202  013246  100401                  BMI     .+4             ;BRANCH IF CHAR PRESENT
3203  013250  104001                  HLT+1                   ;CHARACTER PRESENT GONE!
3204
3205  013252  032701  070000          BIT     #070000,R1      ;CHECK RECEIVER ERRORS
3206  013256  001401                  BEQ     .+4             ;BRANCH IF OK
3207  013260  104001                  HLT+1                   ;RECEIVER  ERROR
3208                                                          ;R1=CONTENTS OF RBUF
3209                                                          ;BIT14=UART OVERRUN
3210                                                          ;BIT13=FRAMMING ERROR
3211                                                          ;BIT12=PARITY ERROR
3212
3213  013262  010102                  MOV     R1,     R2      ;PUT LINE # IN R2
3214  013264  000302                  SWAB    R2
3215  013266  042702  177760          BIC     #177760,R2
3216  013272  120102                  CMPB    R1,     R2      ;CHECK DATA (=LINE#)
3217  013274  001401                  BEQ     .+4             ;BRANCH IF OK
3218  013276  104001                  HLT+1                   ;WRONG DATA RECEIVED
3219                                                          ;R1=FI/FO DATA
3220                                                          ;(DATA SHOULD=LINE#)
3221  013300  005300          22$:    DEC     R0              ;COUNT CHARACTERS
3222  013302  001403                  BEQ     24$             ;BRANCH WHEN DONE
3223  013304  105305          23$:    DECB    R5              ;SHORT WAIT LOOP - GIVE FI/FO TIME
3224  013306  001376                  BNE     23$
3225  013310  000754                  BR      21$             ;GO READ ANOTHER
3226
3227  013312  017701  165672  24$:    MOV     @CSR,   R1      ;SAVE CSR
3228  013316  022701  100605          CMP     #100605,R1      ;CHECK THAT FI/FO OVERRUN CLEARED
3229  013322  001401                  BEQ     .+4             ;BRANCH IF OK
3230  013324  104001                  HLT+1                   ;FI/FO OVERRUN DIDN'T CLR
3231                                                          ;OR SOMEOTHER CSR PROBLEM
3232                                                          ;R1=CONTENTS OF CSR
3233
```

```
                        ;*****************************************************************
                        ;TEST 600:      TEST THAT FI FO O.ERRUN INTERRUPT
                        ;               DOESN'T OCCUR WHEN THE PROCESSOR IS AT LEVEL 5
                        ;*****************************************************************

                        1600:   MOV     #340,   @#PS    ;CLEAR PSW
                                MOV     #340,   @#PS    ;SET PROCESSOR TO LEVEL 5
                                MOV     #RCVR,  @#RCVR.VEC ;SET UP RECEIVER INTERRUPT VEC
                                MOV     #5,     @#RCVR.CSR
                                MOV     #100,   @RCVR.CSR ;SET STATUS ENABLE
                                MOV     @RCVR.CSR,R0    ;MOVE CSR
                                BIT     #100005,R0      ;CHECK CSR
                                BEQ     .+4
                                JSR     R5,ERROR        ;CSR ERROR
                                .WORD   RCVR.CSR        ;R0=CONTENTS OF CSR
                                .WORD   LINE#           ;SEND LINE #
                        336:    TSTB    @#TPB           ;WAIT FOR TRANSMITTER READY
                                BPL     336             ;SEND LINE #
                                MOVB    LINE#,@#TPB     ;WAIT FOR FI FO OVERRUN
                        338:    BIT     #BIT13,@#RCVR.CSR
                                BEQ     338             ;SKIP ISR
                                MOV     #ISR,   @#4     ;SAVE ISR
                                MOV     #ERRUPT,@#4+2   ;INTERRUPT OCCURRED AT LEVEL 5
                                MOVB    @#RCVR.BUF,R0   ;READ ONE CHARACTER
                                MOV     #RCVR.CSR,R0    ;RESET RECEIVER ADDRESS
                                RTS     R5              ;RETURN
                        ;
                        335:    MOV     #2,     R0      ;SET UP COUNTER - 2 CHARACTERS
                        340:    TSTB    @#RCVR.BUF      ;READ ONE CHARACTER
                                BEQ     .+4             ;BRANCH IF CHARACTER PRESENT
                                DEC     R0
                                BNE     340
                        ;
                                MOV     #170000,R1      ;CHECK ERRORS
                                BIT     R1,     R0
                                BEQ     .+4
                                JSR     R5,ERROR
                        ;
                                MOV     R0,     R2      ;DUP DATA
                                BIC     #177760,R2      ;CLR ALL BUT LINE #
                                CMP     R0,     R2      ;CHECK DATA
                                BEQ     .+4
                                JSR     R5,ERROR
                        ;
                                DEC     R0              ;COUNT CHARACTERS
                                BMI     345             ;BRANCH IF DONE
                        355:    TSTB    @#RCVR          ;SHORT WAIT LOOP
                                BPL     355
                                BR      340             ;GO READ ANOTHER CHARACTER
                        ;
                        345:    MOV     @#ISR,  R1      ;SAVE ISR
                                CMP     #ERRUPT,@#4     ;CHECK THAT FI FO O.ERRUN CLEARED
                                BEQ     .+4             ;BRANCH IF OK
```

```
013540  104001                  HLT+1                   ;FI-FO OVERRUN DIDN'T CLR
                                                        ;OR SOMEOTHER CSR PROBLEM
                                                        ;R1=CONTENTS OF CSR

                        ;*****************************************************************
                        ;TEST 600:      TEST THAT FI FO OVERRUN INTERRUPT
                        ;               OCCURS WHEN PROCESSOR IS AT LEVEL 4
                        ;*****************************************************************

[illegible]  [illegible]  [illegible]  [illegible]  TST600:  BIC     #340,   @#PS    ;CLEAR PROCESSOR LEVEL
[illegible]  [illegible]  [illegible]  [illegible]          BIS     #200,   @#PS    ;SET PROCESSOR TO LEVEL 4
[illegible]  [illegible]  [illegible]  165434               MOV     #400$,  @RCVEC  ;SET UP RECEIVER INTERRUPT VEC
[illegible]  [illegible]  [illegible]  [illegible]          MOV     @CSR,   R1      ;SAVE CSR
[illegible]  [illegible]  [illegible]  [illegible]          CMP     #110605,R1      ;CHECK CSR
                                                            BEQ     .+4
                                                            HLT+1

[illegible]  [illegible]  [illegible]  [illegible]  10$:    MOV     #'#,    @#TPB   ;SEND LINE #
[illegible]  [illegible]  [illegible]  [illegible]          TSTB    @#TCSR          ;WAIT FOR TRANSMITTER READY
[illegible]  [illegible]  [illegible]  165400               MOV     #TBUF,  @#TPB   ;SEND LINE #
[illegible]  [illegible]  [illegible]  [illegible]  41$:    MOV     @#RCSR, R1      ;SAVE CSR
[illegible]  [illegible]  [illegible]  [illegible]          BIT     #40,    R1      ;WAIT FOR FI-FO OVERRUN
                                                            BEQ     41$
                                                            HLT+1                   ;INTERUPT DIDN'T OCCURE WHEN OVERRUN SET
                                                                                    ;R1 = CONTENTS OF CSR
[illegible]  [illegible]  000010  165350                    BR      45$             ;CLEAR MOS
[illegible]  [illegible]  [illegible]                       45$                     ;SKIP TO THE END

013642  017701  165342  42$:    MOV     @CSR,   R1      ;SAVE CSR
[illegible]  [illegible]  [illegible]                       CMP     #130605,R1      ;CHECK CSR
[illegible]  [illegible]  [illegible]                       BEQ     .+4             ;BRANCH IF OK
[illegible]  [illegible]  [illegible]                       HLT+1                   ;CSR ERROR
                                                                                    ;R1 = CONTENTS OF CSR
[illegible]  [illegible]  000101  [illegible]               MOV     #101,   R0      ;INIT LOOP COUNTER = 65 CHARACTERS
[illegible]  [illegible]  [illegible]  165334  43$:        MOV     @RBUF,  R1      ;READ ONE CHARACTER
[illegible]  [illegible]  [illegible]  [illegible]          BIT     #200,   R1      ;BRANCH IF CHARACTER PRESENT
                                                            BEQ     .+4
                                                            HLT+1

[illegible]  [illegible]  [illegible]  070000               BIT     #70000, R1      ;CHECK ERRORS
[illegible]  [illegible]  [illegible]  [illegible]          BEQ     .+4
                                                            HLT+1

[illegible]  [illegible]  [illegible]  [illegible]          MOV     R1,     R2      ;DUP DATA
                                                            BIC     #377,   R2
[illegible]  [illegible]  [illegible]  177760               SUB     #177760,R2      ;CLR AL BUT LINE #
[illegible]  [illegible]  [illegible]  [illegible]          CMPB    R1,     R2      ;CHECK DATA
                                                            BEQ     .+4
                                                            HLT+1

[illegible]  [illegible]  [illegible]  165264               MOV     @#RCSR, R1      ;SAVE CSR
                                                            BIC     #177400,R0
                                                            44$:
[illegible]  [illegible]  000100  [illegible]               CMP     #100,   R0      ;CHECK FOR FIRST CHAR READ
                                                            BEQ     44$             ;SKIP CSR CHECK ON FIRST CHAR
[illegible]  [illegible]  [illegible]  [illegible]          CMP     #130605,R1      ;CHECK CSR
```

```
014042  013746  000436                          BEQ     .+4
014044  013746  104001                          HLT+1

014046  013746  000746                          BR      43$

014050  013746  000436          110435  44$:    CMP     #110435,R1      ;CHECK CSR
014054  001402                                  BEQ     .+4             ;BRANCH IF OK
014056  104001                                  HLT+1                   ;DONE DIDN'T GO AWAY
                                                                        ;OR OTHER CSR ERROR

014060  013746  000750          165286          MOV     @RBUF,  R1      ;READ A CHARACTER
014064  100402                                  BPL     .+4             ;BRANCH IF NO CHAR. PRES
014066  104001                                  HLT+1                   ;CHAR. PRESENT!

014070  013746  013777  000060  165266          MOV     RCVLVL, @RCVVEC
014076  013777  000340  165316                  MOV     #340,   @RCVLVL
014104  012716  014012                          MOV     #45$,   (SP)    ;RESET RETURN ADDRESS ON STACK
                                                                        ;RESTORE PSW
014110  005037  165176          45$:    CLR     @TCR
014114  005037  165164                  CLR     @RCSR
014120  042737  000340  177776          BIC     #340,   @#PS    ;LOWER PROCESSOR STATUS

014130  104400                          SCOPE

                                ;*******************************************************************
                                ;TEST 61:       TEST THAT UART OVERRUN IS DETECTED ON ALL LINES
                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E3, E1; M7279; M7280
                                ;*******************************************************************

014132  012777  000414  165150  TST61:  MOV     #414,   @RCSR   ;BIT2 = MAINTENANCE
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT8 = TRANS SCAN ENABLE
014140  032777  000020  165142  10$:    BIT     #BIT4,  @RCSR   ;WAIT FOR MOS TO CLEAR
014146  001374                          BNE     10$
014150  005001                          CLR     R1              ;SET UP LINE COUNTER
014152  012777  000001  165134          MOV     #1,     @TCR    ;TRANS CONTROL = LINE 0
014160  013702  165124          LOP61:  MOV     @RCSR,  R2      ;WAIT FOR TRANS READY
014164  100375                          BPL     LOP61
014166  012777  000001  165122          MOV     #1,     @TBUF   ;SEND #1
014174  013702  165110          25$:    MOV     @RCSR,  R2      ;WAIT FOR TRANS READY
014200  100375                          BPL     25$
014202  012777  000002  165106          MOV     #2,     @TBUF   ;SEND #2
014210  013700  001302                  MOV     @TIMER, R0      ;SET UP TIMER
014214  005300                  35$:    DEC     R0              ;SHORT TIME LOOP
014216  001376                          BNE     35$
014220  013702  165064                  MOV     @RCSR,  R2      ;SAVE CSR FOR THE RECORD
014224  000240                          NOP                     ;NOP FOR TIMING
014226  005300                          DEC     R0              ;TIMER COUNT
014230  001370                          BNE     35$
014232  012777  000100  165052          MOV     #100,   @RCSR   ;SET RECEIVER ENABLE
014240  013702  165046                  MOV     @RCSR,  R2      ;SAVE CSR
014244  032702  000001                  BIT     #BIT0,  R2      ;CHECK RECEIVER ENABLE
014250  001402                          BEQ     .+4             ;BRANCH IF OK
014252  104002                          HLT+2                   ;RECEIVER ENABLE FAILED TO SET
                                                                ;R1 = LINE #
```

```
                                                        ;R2 = CONTENTS OF CSR
                                                        ;WAIT FOR DONE
014152  105737  165032          4$:   TSTB    @CSR
014156  100375                BPL     4$
014160  016102  165026              MOV     @RBUF,  R2  ;READ FIFO
014164  100401                BMI     .+4             ;BRANCH  IF CHARACTER PRESENT
014166  104002                HLT+2                   ;NO CHARACTER PRESENT
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
014170  032702  040000              BIT     #BIT14, R2  ;CHECK FOR UART OVERRUN
014174  001403                BEQ     .+4             ;BRANCH IF OK
014176  104002                HLT+2                   ;UART OVERRUN MISSING
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
014200  032702  000002              BIT     #2,     R2  ;CHECK THE DATA
014204  001403                BEQ     .+4             ;BRANCH IF OK
014206  104002                HLT+2                   ;DATA ERROR - 3RD CHAR OVERRUNS END
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
                                                        ;LOOP CNTR
                                                        ;MASK ALL BIT LINE #, ERROR BITS
                                                        ;CHECK LINE #, ERRORS
                                                        ;BRANCH IF OK
                                                        ;LINE # OR OTHER RBUF ERROR
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
                                                        ;CHECK FOR MORE DATA
                                                        ;BRANCH IF OK
                                                        ;EXTRA DATA IN FIFO!
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
                        7$:   BR      7$

                        9$:   DEC     @CSR            ;CLEAR RECEIVER ENABLE
                              MOV     @CSR,   R2      ;SAVE CSR
                              CMP     #100404,R2      ;CHECK CSR
                              BEQ     .+4             ;BRANCH IF OK
                              HLT+2                   ;CSR ERROR
                                                        ;R1 = LINE NUMBER
                                                        ;R2 = CONTENTS OF CSR
                              INC     R1              ;COUNT LINES
                              ASL     @TCR            ;GO TO NEXT LINE
                              BCC     LOP61           ;BRANCH BACK IF MORE LINES

014270  005077  164714        CLR     @CSR

014274  104400                SCOPE


                        ;*****************************************************************
                        ;TEST 62:       TEST BITS OF BCSR FOR READ WRITE CAPABILITY
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5, E1, (D2-3) E16, E8, E19, E35
                        ;*****************************************************************

014276  012777  002010  164704  TST62:  MOV     #002010,@BCSR   ;SET CSR
014304  012777  177777  164702          MOV     #177777,@BCSR   ;SET ALL BITS OF BCSR
```

```
 3458  014312  017701  164576          MOV     @BCSR,  R1      ;CHECK AND SAVE BCSR
 3459  014316  022701  177777          CMP     #177777,R1      ;CHECK THAT ALL THE BITS ARE SET
 3460  014322  001401                  BEQ     .+4             ;BRANCH IF OK
 3461  014324  104001                  HLT+1                   ;BIT(S) OF BCSR FAILED TO SET
 3462
 3463  014326  005077  164552          CLR     @BCSR           ;CLEAR BCSR
 3464  014332  017701  164546          MOV     @BCSR,  R1      ;CHECK THAT IT CLEARED AND SAVE
 3465  014336  001401                  BEQ     .+4             ;BRANCH IF CLR
 3466  014340  104001                  HLT+1                   ;BIT(S) OF BCSR FAILED TO CLEAR
 3467
 3468  014342  104400                  SCOPE

                                ;*****************************************************************
                                ;TEST 63:       TEST THAT LINE0 CAN TRANSMIT AND RECEIVE A BREAK
                                ;       ALSO CHECKS FRAMING ERROR(RBUF BIT13)
                                ;       ALSO CHECKS PARITY ERROR(RBUF BIT12)
                                ;       IF ODD PARITY IS SELECTED
                                ;PROBABLE FAULTY LOGIC:  M7205 (D2-2) E5, E1, (D2-3) E16, E2, E19, E3E
                                ;*****************************************************************

 3469  014344  004737  015536  TST63:  JSR     PC,     @#INITA ;INITIALIZE
                                                                ;BIT2 = MAINTENANCE
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT10= R/W BCSR
                                                                ;WAIT FOR MOS TO CLEAR
                                                                ;BIT0 = RECEIVER ENABLE
 3470  014350  012777  000001  164536          MOV     #1,     @BCSR   ;SEND BREAKS, LINE 0
 3471  014356  013700  001302                  MOV     TIMER,  R0      ;SET UP TIMER
 3472  014362  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
 3473  014364  001376                  BNE     1$
 3474  014366  017701  164516          MOV     @CSR,   R1      ;SAVE CSR
 3475  014372  105701                  TSTB    R1              ;CHECK FOR DONE
 3476  014374  100404                  BMI     2$              ;BRANCH WHEN DONE
 3477  014376  005200                  INC     R0              ;WAIT A WHILE
 3478  014400  001370                  BNE     1$
 3479  014402  104001                  HLT+1                   ;DONE NEVER CAME UP
                                                                ;R1=CONTENTS OF CSR
 3480  014404  000430                  BR      4$
 3481  014406  022701  002205          CMP     #2205,  R1      ;CHECK REST OF CSR
 3482  014412  001401                  BEQ     .+4             ;BRANCH IF OK
 3483  014414  104001                  HLT+1                   ;CSR ERROR
                                                                ;R1=CONTENTS OF CSR
 3484
 3485  014416  017701  164570          MOV     @RBUF,  R1      ;GET DATA FROM FIFO
 3486  014422  100401                  BMI     .+4             ;BRANCH IF OK
 3487  014424  104001                  HLT+1                   ;CHARACTER PRESENT NOT UP
                                                                ;R1=CONTENTS OF RBUF
 3488
 3489  014426  032701  020000          BIT     #020000,R1      ;CHECK FOR FRAMING ERROR
 3490  014432  001001                  BNE     .+4             ;BRANCH IF OK
 3491  014434  104001                  HLT+1                   ;FRAMING ERROR NOT UP
                                                                ;R1=CONTENTS OF RBUF
 3492  014436  133737  001310  001276  BITB    DJPAR,  PARITY  ;CHECK ODD PARITY FLAG
 3493  014444  001404                  BEQ     3$              ;BRANCH IF NOT
 3494  014446  032701  010000          BIT     #010000,R1      ;CHECK PARITY ERROR
 3495  014452  001035                  BNE     4$              ;
```

```
014454  104001                          HLT+1                   ;ODD PARITY SHOULD CAUSE PARITY ERROR
                                                                ;R1=CONTENTS OF RBUF

014456  032701  010000          35:     BIT     #010000,R1      ;CHECK PARITY ERROR
014462  001401                          BEQ     .+4             ;BRANCH IF OK
014464  104001                          HLT+1                   ;EVEN PARITY OR NO PARITY
                                                                ;SHOULDN'T CAUSE PARITY ERROR
                                                                ;R1=CONTENTS OF RBUF
014466  032701  040000          45:     BIT     #040000,R1      ;CHECK UART OVERRUN
014472  001401                          BEQ     .+4             ;BRANCH IF OK
014474  104001                          HLT+1                   ;UART OVERRUN SET!!
                                                                ;R1=CONTENTS OF RBUF
014476  032701  007400                  BIT     #007400,R1      ;CHECK LINE #
014502  001401                          BEQ     .+4             ;BRANCH IF OK
014504  104001                          HLT+1                   ;WRONG LINE# IN FIFO
                                                                ;R1=CONTENTS OF RBUF
014506  105701                          TSTB    R1              ;CHECK DATA
014510  001401                          BEQ     .+4             ;BRANCH IF OK
014512  104001                          HLT+1                   ;WRONG DATA RECEIVED
                                                                ;R1=CONTENTS OF RBUF
014514  017701  164472                  MOV     @RBUF, R1       ;READ FIFO AGAIN
014520  001401                          BEQ     .+4             ;BRANCH IF OK
014522  104001                          HLT+1                   ;EXTRA CHAR IN FIFO
                                                                ;R1 = CONTENTS OF RBUF
014524  005077  164464                  CLR     @RCSR           ;CLEAR BREAK CONTROL REG
014530  005000          55:     DEC     R0              ;SHORT WAIT LOOP-REGISTER = NBR
014532  001376                  BNE     55
014534  104400                  SCOPE


                        ;*****************************************************************
                        ;TEST 64:      TEST THAT EACH LINE CAN TRANSMIT AND RECEIVE A BREAK
                        ;              ALSO CHECKS FRAMING ERROR,RBUF BIT 13
                        ;              ALSO CHECKS PARITY ERROR,RBUF BIT 12
                        ;              ALL 000 CHARACTERS ARE RE-ENTERED
                        ;PROBABLE FAULTY LOGIC:  M265 M26-07 E5, E1, M26-3  E15, E3, E19, E3E
                        ;*****************************************************************

014536  004737  015536          TST64:  JSR     R0,     @#INITA ;INITIALIZE
                                                                ;BIT 2 = MAINTENANCE
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT10= RAISE RCSR
                                                                ;BIT11= FOR MOS RCSR CLEAR
                                                                ;BIT12 = RECEIVER ENABLE
014542  005001                  CLR     R1              ;SET UP LINE COUNTER
014544  012704  000001          MOV     #1,     R4      ;SET UP LINE# MARKER
014550  013705  001302  LOP64:  MOV     @#TIMER, R5     ;SET UP TIMER
014554  060477  164434          ADD     R4,     @RCSR   ;SET BREAK CONTROL BIT, LINE # IN R1
014560  105305          1$:     DECB    R5              ;SHORT WAIT LOOP
014562  001376                  BNE     1$
014564  017702  164420          MOV     @RCSR,  R2      ;SAVE CSR
014570  105702                  TSTB    R2              ;CHECK FOR DONE
014572  100004                  BPL     2$              ;BRANCH WHEN DONE
014574  005202                  INC     R2              ;WAIT A WHILE
```

```
014676  001377          BNE     1$
014600  104002          HLT+2                   ;DONE NEVER CAME UP
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF CSR

014602  022705  002205  3$:     CMP     #2205,  R5      ;CHECK REST OF CSR
014606  001402          BEQ     .+4             ;BRANCH IF OK
014610  104003          HLT+3                   ;CSR ERROR
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF CSR

014612  016705  164372  4$:     MOV     RBUF,   R5      ;GET DATA FROM FIFO
014616  100402          BPL     .+4             ;BRANCH IF OK
014620  104004          HLT+4                   ;CHARACTER PRESENT NOT UP
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF RBUF

014622  032705  020000  5$:     BIT     #020000,R5      ;CHECK FOR FRAMING ERROR
014626  001402          BEQ     .+4             ;BRANCH IF OK
014630  104002          HLT+2                   ;FRAMING ERROR NOT UP
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF RBUF

014632  010203          MOV     R2,     R3      ;GET LINE #
014634  000303          SWAB    R3              ;
014636  133763  001310  001276  BITB    DIPAR,PARITY(3) ;CHECK ODD PARITY FLAG
014644  001402          BEQ     3$              ;BRANCH IF NOT
014646  032702  010000          BIT     #010000,R2      ;CHECK PARITY ERROR
014652  001005          BNE     4$              ;
014654  104003          HLT+2                   ;ODD PARITY SHOULD CAUSE PARITY ERROR
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF RBUF

014656  032702  010000  3$:     BIT     #010000,R2      ;CHECK PARITY ERROR
014662  001401          BEQ     4$              ;BRANCH IF OK
014664  104003          HLT+2                   ;EVEN PARITY OR NO PARITY
                                                ;SHOULDN'T CAUSE PARITY ERROR
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF RBUF

014666  032702  040000  4$:     BIT     #040000,R2      ;CHECK UART OVERRUN
014672  001402          BEQ     .+4             ;BRANCH IF OK
014674  104002          HLT+2                   ;UART OVERRUN SET!!
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF RBUF

014676  010203          MOV     R2,     R3      ;DUP DATA
014700  000303          SWAB    R3              ;LINE # IN LOW BYTE
014702  042703  177760  BIC     #177760,R3      ;MASK ALL BUT LINE #
014706  020103          CMP     R1,     R3      ;CHECK FOR RIGHT LINE #
014710  001402          BEQ     .+4             ;BRANCH IF OK
014712  104002          HLT+2                   ;WRONG LINE# IN FIFO
                                                ;R1 = LINE #
                                                ;R2 = CONTENTS OF RBUF

014714  105702          TSTB    R2              ;CHECK DATA
```

```
                                                BEQ     .+4             ;BRANCH IF OK
                                                HLT+2                   ;WRONG DATA RECEIVED
                                                                        ;R1 = LINE #
                                                                        ;R2 = CONTENTS OF RBUF
                                                MOV     @RBUF,  R2      ;READ FI/FO AGAIN
                                                BPL     .+4             ;BRANCH IF OK
                                                HLT+2                   ;EXTRA CHAR IN FI/FO
                                                                        ;R1 = LINE #
                                                                        ;R2 = CONTENTS OF RBUF
                                                CLR     @BCSR           ;CLEAR BREAK CONTROL REG
                                                INC     R1              ;COUNT LINES
                                                ASL     R4              ;UPDATE LINE MARKER
                                                BCC     LOP64           ;BRANCH IF MORE LINES

                                                CLR     @CSR            ;CLEAR CSR
                                                SCOPE

                                                MOV     #2.     TIMES

                        ;*******************************************************************************
                        ;TEST 65:       TEST THAT RESET CLEARS ALL BUFFERS
                        ;       NOTE:   THE FI/FO BUFFER IS NOT COMPLETELY CLEARED
                        ;               BY RESET; ONLY CHARACTER PRESENT IS CLEARED.
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-8) E13
                        ;*******************************************************************************

                        TST65:  BIS     #340,@#PS       ;SET PROCESSOR TO LEVEL 7
                                MOV     #177777,@TCR    ;SET ALL TCR BITS
                                MOV     #052507,@CSR    ;SET ALL R/W BITS OF CSR
                                MOV     #177777,@BCSR   ;SET ALL BREAK CONTROL BITS (SEND BREAKS)
                        ;NOTE:  ALL LINES SHOULD BE SENDING BREAKS, BUT NONE SHOULD BE RECEIVING
                        ;         BECAUSE THE HALF DUPLEX BIT IS SET
                                MOV     #177777,@TBUF   ;LOADING TRANS BUFF WHEN BREAK BIT SET
                                                        ;  SHOULD DO NOTHING
                                RESET                   ;CLEAR THE WORLD
                                MOV     CSR,1$          ;CHECK CSR AND SAVE
                                MOV     @(PC)+,R1
                        1$:     000000
                                BEQ     .+4
                                HLT+1

                                MOV     @TCR,R1         ;CHECK TCR AND SAVE
                                BEQ     .+4
                                HLT+1

                                MOV     @BCSR,R1        ;CHECK BCSR AND SAVE
                                BEQ     .+4
                                HLT+1

                                MOV     @RBUF,R1        ;CHECK RBUF AND SAVE
                                BPL     .+4
                                HLT+1

                                MOV     @TBUF,R1        ;CHECK TBUF AND SAVE
                                BEQ     .+4
                                HLT+1
```

```
                                        ;*****************************************************************
                                        ;TEST 66:       SEND A BINARY COUNT PATTERN ON EACH LINE
                                        ;PROBABLE FAULTY LOGIC:  COULD BE ALMOST ANYWHERE!
                                        ;*****************************************************************

015102  104400                          SCOPE

015104  005001          TST66:  CLR     R1                      ;SET UP LINE COUNTER
015106  012703  100000          MOV     #100000,R3              ;SET UP RCV DATA
015112  005004          CLR     R4                      ;SET UP TRANS DATA
015114  042737  000340  177776  BIC     #340,   @#PS            ;CLEAR PROCESSOR PRIORITY
015122  052737  000200  177776  BIS     #200,   @#PS            ;SET PRIORITY TO 4
015130  012700  000001          MOV     #1,     R0              ;SET UP LINE MARKER
015134  012777  000010  164046  MOV     #10,@DCSR               ;CLEAR MOS
015142  012777  010505  164040  MOV     #010505,@DCSR           ;BIT0 = RECEIVE ENABLE
                                                                ;BIT2 = MAINTENANCE
                                                                ;BIT6 = RECEIVER INTERRUPT ENB
                                                                ;BIT8 = TRANS SCAN ENABLE
                                                                ;BIT12= STATUS ENABLE
015150  032777  000020  164032  10$:    BIT     #BIT4,  @DCSR   ;WAIT FOR MOS TO CLEAR
015156  001374          BNE     10$
015160  012777  015274  164032          MOV     #ISR66,@RCVVEC  ;SET UP RECEIVER INTERRUPT VECTOR
015166  012777  000240  164026          MOV     #240,   @RCVLVL
015174  010077  164014          1$:     MOV     R0,     @TCR    ;TRANS CONTROL, ONE LINE AT A TIME
015200  005777  164004          2$:     TST     @DCSR           ;WAIT FOR TRANS READY
015204  100375          BPL     2$
015206  010477  164004          MOV     R4,     @TBUF   ;SEND DATA
015212  105204          INCB    R4                      ;BINARY COUNT
015214  001371          BNE     2$
015216  105703          3$:     TSTB    R3                      ;WAIT FOR RECEIVER DONE
015220  001376          BNE     3$
015222  017702  163762          MOV     @DCSR,  R2              ;SAVE CSR
015226  022702  110505          CMP     #110505,R2              ;CHECK CSR
015232  001401          BEQ     .+4                     ;BRANCH IF OK
015234  104002          HLT+2                           ;CSR ERROR
                                                        ;R1=LINE #
                                                        ;R2=CONTENTS OF CSR
015236  017702  163750          MOV     @RBUF,  R2              ;CHECK CHARACTER PRESENT
015242  100001          BPL     .+4                     ;BRANCH IF OK
015244  104002          HLT+2                           ;CHARACTER PRESENT SET!!
                                                        ;R1=LINE #
                                                        ;R2=CONTENTS OF CSR
015246  062703  000400          ADD     #400,   R3              ;UPDATE LINE # IN EXPECTED DATA
015252  005201          INC     R1                      ;UPDATE LINE #
015254  032701  000003          BIT     #3,     R1              ;CHECK FOR FOURTH LINE
015260  001002          BNE     4$                      ;BRANCH IF NOT
015262  005237  001306          INC     DJLEN                   ;MOVE CHARACTER LENGTH POINTER
015266  006300          4$:     ASL     R0                      ;UPDATE LINE MARKER
015270  103341          BCC     1$                      ;BRANCH IF MORE
015272  000417          BR      END66                   ;SKIP ISR

015274  017702  163712  ISR66:  MOV     @RBUF,  R2              ;READ FIRST DATA
015300  100401          BMI     11$                     ;BRANCH IF CHARACTER PRESENT
015302  104003          HLT+3                           ;INTERRUPT BUT NO CHAR PRESENT
```

```
3738                                                                    ;R1=LINE #
3739                                                                    ;R2=CONTENTS OF RBUF
3740                                                                    ;R3=EXPECTED DATA
3741  015304  020203                    11$:    CMP     R2,     R3    ;CHECK THE DATA
3742  015306  001401                            BEQ     .+4           ;BRANCH IF OK
3743  015310  104003                            HLT+3                 ;DATA ERROR
3744                                                                    ;R1=LINE #
3745                                                                    ;R2=CONTENTS OF RBUF
3746                                                                    ;R3=EXPECTED DATA
3747  015312  105203                            INCB    R3            ;UPDATE EXPECTED DATA
3748  015314  147703  163766                    BICB    @DJLEN, R3    ;MASK CHARACTER LENGTH
3749  015320  001403                            BEQ     12$           ;BRANCH OUT IF DATA=0
3750  015322  017702  163664                    MOV     @RBUF,  R2    ;READ MORE DATA
3751  015326  100766                            BMI     11$           ;BRANCH IF MORE
3752  015330  000002                    12$:    RTI                   ;RETURN
3753
3754  015332  162737  000004  001306  END66:  SUB     #4,     DJLEN ;RESET CHAR LENGTH POINTER
3755  015340  013777  001222  163652          MOV     RCVLVL, @RCVVEC ;RESTORE RECEIVER INT. VEC
3756  015346  012777  000004  163646          MOV     #IOT,   @RCVLVL
3757  015354  005077  163634                  CLR     @TCR          ;CLEAR TCR
3758  015360  005077  163624                  CLR     @CSR          ;CLEAR CSR
3759  015364  104400                          SCOPE
3760
3761  015366  012737  000020  015772          MOV     #20,    TIMES
3762  015374  023737  001304  CC1234          CMP     DJUUT.UNITS   ;CHECK FOR LAST UNIT
3763  015402  002002                          BGE     DONE          ;BRANCH IF LAST UNIT
3764  015404  000137  002302                  JMP     RESTAR        ;JUMP IF NOT
3765
3766  015410                            DONE:
3767  015410  004737  017640                  JSR     PC,     KBDINT
3768  015414  062737  000001  001206          ADD     #1,PCNT+2     ;ADD 1 TO THE PASS COUNT
3769  015422  005537  001204                  ADC     PCNT          ;MAKE IT DOUBLE PREC.
3770  015426  032777  002000  163662          BIT     #SW10,@SWR    ;RING THE BELL?
3771  015434  001004                          BNE     1$            ;NO!
3772  015436  000004  000007                  TYPE    .BELL         ;RING THE BELL
3773  015442  000004  000177                  TYPE    ,177          ;TYPE A FILLER FOR 11/05
3774  015446  005046                    1$:   CLR     -(6)          ;CLEAR TRACE TRAP
3775  015450  032777  010000  163640          BIT     #SW12,@SWR    ;RUN WITH TRT?
3776  015456  001010                          BNE     2$            ;SKIP T BIT
3777  015460  005137  015532                  COM     .TBIT         ;COMPLIMENT FLAG
3778  015464  100005                          BPL     2$            ;SKIP IF PLUS
3779  015466  052716  000020                  BIS     #20,(6)       ;SET TRACE TRAP
3780  015472  012746  015526                  MOV     #3$,-(6)      ;JUMP TO START OF TEST
3781  015476  000002                          RTI                   ;RETURN
3782  015500  012746  015506          2$:     MOV     #4$,-(6)      ;JUMP TO START OF TEST
3783  015504  000002                          RTI                   ;RETURN
3784  015506  013700  000042          4$:     MOV     @#42,R0       ;GET MONITOR ADDRESS
3785  015512  001405                          BEQ     3$            ;IF NONE
3786  015514  000005                          RESET                 ;RESET AND
3787          015516                          $ENDAD = .
3788  015516  004710                          JSR     7,(0)         ;GO TO MONITOR
3789  015520  000240                          NOP                   ;SAVE ROOM
3790  015522  000240                          NOP                   ;FOR
3791  015524  000240                          NOP                   ;ACT11
3792  015526  000137  002302          3$:     JMP     RESTAR        ;RETURN
3793
```

```
 3794  015532  000000                .TBIT:  0                       ;T BIT FLAG
 3795
 3796  015534  000002                ESRT:   RTI                     ;RETURN FROM TRACE TRAP
                                      ;
                                      ;   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
                                      ;BIT09 = MASTER XMTR SCAN ENB
                                      ;BIT10 = R/W BCSR
                                      ;
                                      ;WAIT FOR MOS TO CLEAR
                                      ;SET:
                                      ;BIT00 = RECEIVER ENABLE
                                      ;
                                                              SET
 3817  015536  012777  002014  163444 INITA:  MOV     #2014,@CSR      ;BIT(S)2,3,10 THEN 0
 3818  015544  000413                        BR      INITR
 3819
 3820  015546  012777  000514  163434 INITB:  MOV     #514,@CSR       ;BIT(S)2,3,6,8 THEN 0
 3821  015554  000407                        BR      INITR
 3822
 3823  015556  012777  000416  163424 INITC:  MOV     #416,@CSR       ;BIT(S)1,2,3,8 THEN 0
 3824  015564  000403                        BR      INITR
 3825
 3826  015566  012777  000414  163414 INITD:  MOV     #414,@CSR       ;BIT(S)2,3,8 THEN 0
 3827
 3828  015574  005000                INITR:  CLR     R0
 3829
 3830  015576  005200                1$:     INC     R0              ;ANTIHANG
 3831  015600  001004                        BNE     2$              ;ROUTINE
 3832
 3833  015602  011600                        MOV     (SP),R0         ;RECORD SUBROUTINE CALL RETURN
 3834  015604  162700  000002                SUB     #2,R0           ;FORM CALL ADDRESS FOR DISPLAY
 3835
 3836  015610  104000                        HLT                     ;BIT#4 OF DEVICE CSR FAILED TO CLEAR
 3837
 3838  015612  032777  000020  163370 2$:     BIT     #BIT4,@CSR      ;TEST HAS MOS CLEARED
 3839  015620  001366                        BNE     1$              ;NO BRANCHES
 3840                                                                 ;SET:
 3841  015622  052777  000001  163360        BIS     #1,     @CSR    ;BIT00 RECEIVE ENABLE
 3842  015630  000207                        RTS     PC              ;CONTINUE
 3843                                  ;
 3844                                  ;       $SCOPE          SCOPE LOOP HANDLER
 3845
 3846                                  ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
 3847                                  ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
 3848
```

```
                                        ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
                                        ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
                                
015632  004737  017640          TRAP$:  JSR     PC,     KBDINT
015636  032777  000400  163452          BIT     #SW8,@SWR       ;LOOP ON SPEC. TEST?
015644  001404                          BEQ     1$              ;NO LOOP ON SPEC. TEST
015646  127737  163444  001200          CMPB    @SWR,ICNT       ;ON RIGHT TEST?   *SW7-0*
015654  001434                          BEQ     OVER$           ;NOT RIGHT TEST
015656  032777  040000  163432  1$:     BIT     #SW14,@SWR      ;LOOP ON TEST?
015664  001026                          BNE     KIT$            ;LOOP ON TEST IS SET
015666  032777  004000  163422          BIT     #SW11,@SWR      ;KILL ITERATIONS
015674  001012                          BNE     SVLAD$          ;YES - KILL ITERATIONS
015676  105737  001201                  TSTB    ICNT+1          ;FIRST ONE?
015702  001404                          BEQ     2$              ;BRANCH IF FIRST
015704  123737  015772  001201          CMPB    TIMES,ICNT+1    ;DONE?
015712  001013                          BNE     KIT$            ;BRANCH IF NOT
015714  112737  000001  001201  2$:     MOVB    #1,ICNT+1       ;FIRST ITERATION
015722  105237  001200          SVLAD$: INCB    ICNT            ;COUNT TEST NUMBERS
015726  011637  015770                  MOV     (6),LAD         ;SAVE LOOP ADDRESS
015732  013737  001200  001320          MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
015740  000002                          RTI                     ;RETURN
                                
015742  105237  001201          KIT$:   INCB    ICNT+1          ;INC THE ITERATION COUNT
015746  013737  001200  001320  OVER$:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
015754  005737  015770                  TST     LAD             ;FIRST ONE?
015760  001760                          BEQ     SVLAD$          ;YES
015762  013716  015770                  MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
015766  000002                          RTI                     ;FIXES PS
                                
015770  000000          LAD:    0               ;LOOP ADDRESS
015772  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.
                                        ;
016774  004737  017640          EMT$:   JSR     PC,     KBDINT
016000  032777  002000  163310          BIT     #SW10,@SWR      ;BELL ON ERROR?
016006  001402                          BEQ     1$              ;NO - SKIP
016010  000004  000007                  TYPE    ,BELL           ;RING BELL
016014  005237  001202          1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
016020  032777  020000  163270          BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
016026  001026                          BNE     2$              ;SKIP TYPEOUTS
016030  000004  016034                  TYPE    .+2             ;.ASCIZ <15><12>
016040  011637  016144                  MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
016044  162737  000002  016144          SUB     #2,HLTADR       ;FUDGE ADDRESS
016052  117737  000066  016142          MOVB    @HLTADR,HLTCT$  ;GET HLT ARGUEMENT
016060  013705  016144                  MOV     HLTADR,TTY      ;TYPE HLTADR IN OCTAL
016064  004737  016562                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
016070  000004  016074                  TYPE    .+2             ;.ASCIZ "  "
016100  004737  016146                  JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
016104  005777  163206          2$:     TST     @SWR            ;HALT ON ERROR
016110  100001                          BPL     .+4             ;SKIP IF CONTINUE
016112  000000                          HALT                    ;HALT ON ERROR!
016114  004737  017640                  JSR     PC,KBDINT
016120  032777  001000  163170          BIT     #SW9,@SWR       ;CHECK FOR INHIBIT LOOP ON ERROR
016126  001001                          BNE     .+4             ;SKIP IF LOOP ON ERROR
016130  000002                          RTI                     ;RETURN
016132  105037  001201                  CLRB    ICNT+1          ;CLEAR ITERATION COUNT
016136  000137  015742                  JMP     KIT$            ;LOOP ON TEST UNTIL NO ERRORS
                                        ;
016142  000000          HLTCT$: 0                               ;HLT ARGUEMENT
016144  000000          HLTADR: 0                               ;LAST HLT INSTRUCTION EXECUTED
                                        ;
016146                  ERROR$:
016146  013705  001210                  MOV     CSR,TTY         ;TYPE CSR IN OCTAL
016152  004737  016562                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
016156  042737  007700  016204          BIC     #7700,2$
016164  105337  016142          1$:     DECB    HLTCT$
016170  100411                          BMI     3$
016172  062737  000100  016204          ADD     #100,2$
016200  000004  017241                  TYPE,   SPACE
016204  010005          2$:     MOV     %0,TTY          ;TYPE REGISTER X IN OCTAL
016206  004737  016562                  JSR     %7,PRINTR
016212  000764                          BR      1$
016214  000207          3$:     RTS     PC
```

```
                        ;SUBROUTINE TO SAVE INPUT AS OCTAL NUMBER

                        EADIN: MOV     #1,INHRE
                               JSR     R5,     READ$   ;GO READ TTY UNTIL CR
                               .WORD   INHRE
                               MOV     R1,-(SP)        ;PUSH R1 ON STACK
                               MOV     R2,-(SP)        ;PUSH R2 ON STACK
                               MOV     R3,-(SP)        ;PUSH R3 ON STACK
                               MOVB    (R4)+,  R1
                               BEQ     ES
                               CMPB    R1,     #"P
                               BEQ     1$
                               CMPB    R1,     #"B
                               BNE     ES
                               MOV     #177560,R2      ;CHECK FOR "P"
                               TSTB    (R2)
                               BPL     .-4
                               MOVB    R1,     2(R2)
                               MOV     #100000,R1
                               MOV     #15,    R3
                               CLR     R2
                               ASHC    #3,     R1
                               BIC     #177770,R1
                               BIS     R1,     R2
                               SOB     R3,     .-12
                               MOV     R2,     @(SP)+  ;BRANCH IF BAD DATA
                               BR      ES
                               MOV     (SP)+,  R3      ;RESTORE R3
                               MOV     (SP)+,  R2      ;RESTORE R2
                               MOV     (SP)+,  R1      ;RESTORE R1
                               RTS     PC              ;RETURN

                        ES:    .WORD   0               ;TEMPORARY STORAGE FOR ES

                        1$:    MOV     R3,     -(SP)   ;SAVE R3
                               MOV     #100000,R1      ;SET UP FOR
                               CLR     R2              ;  ASHC
                               MOV     #5,     R3      ;SHIFT 5 TIMES
                               ASHC    #3,     R1      ;SHIFT R1 AND R2
                               BIC     #177770,R1      ;CLEAR ALL BUT LOW 3 BITS
                               BIS     R1,     R2      ;SAVE IN R2
                               SOB     R3,     .-12    ;LOOP
                               MOV     R2,     @(SP)+  ;SAVE IN USER'S WORD
                               BR      ES              ;EXIT

                        4$:    TST     (R5)+           ;SKIP
                               RTS     PC              ;RETURN

                        INHRE: .BLKW   6               ;INPUT HERE
```

```
                                                    ;ECHO IT
                                                    ;CHECK FOR RETURN
                                                    ;LOOP IF NOT RETURN
                                                    ;CHR RETURN, THE IS
                                                    ;  FEED, LINE FEED
                                                    ;RESTORE R3
                                                    ;RETURN
                                                    ;  INPUT AREA
```

```
                                        ;       $OCTAL          OCTAL TYPEOUT ROUTINE

                                        ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
                                        ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
                                        ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMPIS, OR BITYPE MACRO'S.

016656  016737  170101  016720  BITYP$: MOV     #170101,.PR     ;SET BIT FLAG AND 16. CHARACTER COUNT
016662  000404          016720          BR      .PTIT           ;NOW TYPE IT IN BIT FORM
016664  116737  000001  016720  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
016672  000403          016720          BR      .+8             ;SKIP
016674  005037  016720  016720  PRINT$: CLR     .PR             ;SUPPRESS LEADING ZERO'S
016700  116737  177772  016721          MOVB    #-6,.PR+1       ;SET COUNT
016706  010446          016722  .PTIT:  MOV     R4,-(6)         ;SAVE R4
016710  012704  016722                  MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
016714  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
016716  000411                          BR      .PRF            ;ROTATE FIRST BIT
016720  105014          .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
016722  032737  000100  016720          BIT     #100,.PR        ;BIT TYPING MODE?
016730  001004                          BNE     .PRF            ;YES - SKIP 2 ROTATES
016732  006104                          ROL     -(4)            ;ROTATE BIT INTO C
016734  006114                          ROLB    (4)             ;PACK IT
016736  006104                          ROL     -(4)            ;ROTATE BIT INTO C
016740  006114                          ROLB    (4)             ;PACK IT
016742  006104          .PRF:   ROL     -(4)            ;ROTATE BIT INTO C
016744  006114                  ROLB    (4)             ;PACK IT
016746  105714                  TSTB    (4)             ;IS IT ZERO?
016750  001403                  BEQ     .+6             ;SKIP IF SO
016752  105237  016720          INCB    .PR             ;SET FILL SWITCH
016756  100367                  BPL     .PRL            ;CHECK FILL SWITCH
016760  000137  016730          JMP     @#.+10          ;SKIP BYTE
016764  152704  000060          BISB    #'0,R4          ;MAKE INTO ASCII CHAR
016770  105337  016721          DECB    .PR+1           ;ANY MORE COUNT?
016774  001362                  BNE     .PRF            ;REPEAT
016776  012704  016722          MOV     #.PR+2,R4       ;EMPTY BUFFER
017002  112437  000012          MOVB    (4)+,@#.+10     ;SKIP IF NOT
017006  000000                  .WORD   0               ;LOAD 1 CHAR
017010  105337  016720          DECB    .PR             ;ANY TERMINATOR
017014  001371                  BNE     .-12            ;NO
017016  012604                  MOV     (6)+,R4         ;RESTORE R4
017020  000207                  RTS     PC              ;RETURN
017022  000000  000000  000000  .PR:    .BLKW   12      ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
                    017776  000130  PDOWN$: MOV     #ILLUP,@PWRVEC$  ;SET FOR FAST UP
                    000340  000132          MOV     #340,@PWRVEC$+2  ;PRIO:7
                            010046          MOV     R0,-(SP)         ;PUSH R0 ON STACK
                            010146          MOV     R1,-(SP)         ;PUSH R1 ON STACK
                            010246          MOV     R2,-(SP)         ;PUSH R2 ON STACK
                            010346          MOV     R3,-(SP)         ;PUSH R3 ON STACK
                            010446          MOV     R4,-(SP)         ;PUSH R4 ON STACK
                            010546          MOV     R5,-(SP)         ;PUSH R5 ON STACK
                    017776                  MOV     SP,.SAVR6        ;SAVE SP
                    000024  000132          MOV     #PDOWN$,@#24     ;SET UP VECTOR
                    000000                  HALT                     ;WAIT FOR PF

                    017776          PUPS:   MOV     .SAVR6,SP        ;GET SP
                            005001          CLR     R1               ;WAIT LOOP FOR THE ???
                            005201          INC     R1               ;WAIT FOR THE INC
                            001376          BNE     1$               ;OF  WORD
                            012605          MOV     (SP)+,R5         ;POP STACK INTO R5
                            012604          MOV     (SP)+,R4         ;POP STACK INTO R4
                            012603          MOV     (SP)+,R3         ;POP STACK INTO R3
                            012602          MOV     (SP)+,R2         ;POP STACK INTO R2
                            012601          MOV     (SP)+,R1         ;POP STACK INTO R1
                            012600          MOV     (SP)+,R0         ;POP STACK INTO R0
                    000132  000024          MOV     #PDOWN$,@#24     ;SET UP THE POWER DOWN VECTOR
                    000340  000026          MOV     #340,@#26        ;PRIO:7
                                            TYPE    .+2              ;.ASCIZ <15><12>"POWER"
                    177776                  JMP     @#-2             ;JMP TO USER ADDRESS

                                    ILLUP:  HALT                     ;THE POWER UP SEQUENCE WAS STARTED
                    000776                  BR      .-2              ;BEFORE THE POWER DOWN WAS COMPLETE

                    000000          .SAVR6: 0                        ;PUT THE SP HERE
                    000024          PWRVEC: 24.                      ;POWER UP VECTOR
```

```
                                        ;*****************************************************************
                                        ;IOT HANDLER.  REENTERENT ROUTINE TO EITHER TYPE MESSAGES OR
                                        ;       INDICATE A FAULSE INTERUPT OR TRAP.
                                        ;*****************************************************************

017104  000716  001000      IOTRAP: CMP     #1000.  (SP)    ;CHECK RETURN ADDRESS FOR FAULSE TRAP
017110  001401                      BLT     IOT$            ;BRANCH IF "TYPE" COMMAND INTENDED
017112  160716  000004              SUB     #4,     SP)     ;GET VECTOR ADDRESS FROM RETURN ADDRESS
017116  012601                      MOV     (SP)+,  R1      ;PUT IN  R1 FOR TYPING
017120  005726                      TST     (SP)+           ;POP STACK
017122  011602                      MOV     (SP),   R2      ;SAVE RETURN ADDRESS FOR TYPING
017124  104002                      HLT+2                   ;UNEXPECTED INTERJPT OR TRAP
                                                            ;R1 = VECTOR ADDRESS
                                                            ;R2 = RETURN PC
017126  000002                      RTI                     ;CONTINUE THE PROGRAM

                                ;MCALL  $TYPE
                                ;       $TYPE           MESSAGE TYPEOUT ROUTINE

                                ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY.  THE
                                ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
                                ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
                                ;THE ASCII "CHAR", AND 3) "PRINT (<15><12>"MESSAGE") - TYPES
                                ;THE MESSAGE WHICH IS INLINE ASCII.

017130  010546              IOT$:   MOV     TTY,-(6)        ;SAVE TTY
017132  016605  000002              MOV     2(6),TTY        ;GET ADDRESS TO BE TYPED
017136  032705  177400              BIT     #177400,TTY     ;IS IT A TYPE?
017142  001003                      BNE     1$              ;NO
017144  116537  017234              MOV     TTY,.TYPE       ;GET THE CHARACTER
017152  012705  017234              MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
017156  105715              1$:     TSTB    (TTY)           ;TERMINATOR?
017160  001406                      BEQ     2$              ;GET OUT IF SO
017162  112537  177566              MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
017166  105737  177564              TSTB    @#177564        ;IS THE PRINTER READY
017172  100375                      BPL     .-4             ;WAIT UNTIL IT IS
017174  000767                      BR      1$              ;GET THE NEXT CHARACTER
017176  062766  000002      000004  2$:     ADD     #2,     2(6)    ;GET ADDRESS TO BE TYPED
017204  062705  000002              ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
017210  005715                      TST     (TTY)           ;IS IT .+2?
017212  001001                      BNE     3$              ;NO
017214  062705  000002              ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
017220  005715              3$:     TST     (TTY)           ;BACK UP TO EVEN BYTE
017222  116605  000002              MOVB    2(6),TTY        ;RESTORE ADDRESS
017226  012605              4$:     MOV     (6)+,TTY        ;RESTORE TTY
017230  000002                      RTI                     ;RETURN
017232  000000              .TYPE:  0                       ;CHARACTER TYPE LOCATION

017234  005015              RETURN: .ASCIZ  <15><12>
017236  040040              SPACE:  .ASCIZ  "  "
017240  050015  044506  035522  MSGPOR: .ASCIZ  <15><12>"FIRST DJ11 ADDRESS:  "
017246  020124  046104  030461
017254  047040  042040  042040
017262  035440  020072  000040
017270  050015  046505  042103  MSGVEC: .ASCIZ  <15><12>"VECTOR ADDRESS:  "
017276  044503  042104
```

```
051523  020072

047516  020056  MSGNUM: .ASCIZ  15< 12 "NO. OF DJII'S:  "
000040  000000
000103  000040
000040  000000

051412  040564  MSGCON: .ASCIZ  \15< 12 "STANDARD CONFIGURATION"  "
000000  000000

047111  MSGLIN: .ASCIZ  <15  12 "LINES "

        MSGDAS: .ASCIZ  " - "
        MSGLEN: .ASCIZ  <15  12 "CHAR LENGTH:  "

        MSGPAR: .ASCIZ  <15  12 "PARITY:NO. ODD. E.E" :  "


.E.EN

:SUBROUTINE TO AUTOMATICALLY DETERMINE THE NUMBER OF DJII'S IN THE S YSTEM
:       AND WHERE THEIR CONTROL VECTORS ARE.
:THIS ROUTINE IS ONLY USED IF DJ.NUM IS NOT ZERO, AS WHEN THE PROGRAM IS
:       BEING RUN UNDER ACT-11 OR ODT MONITOR.
:NOTE:  SOME OF THE LOGIC MUST BE FUNCTIONAL OR THIS ROUTINE WILL GIVE!

A.TO:   MOV     #DJCSR0,R0      ;START AT DJ RESPONSE BASE ADDRESS
        CLR     R1              ;CLEAR NUMBER OF DJII'S FOUND
        MOV     #10.,R2         ;SET UP TO TEST 8 POSSIBLE DJII'S
1$:     TST     (R0)            ;DOES THIS ADDRESS RESPOND?
        BMI     2$              ;BR IF NO RESPONSE
        INC     R1              ;INCREMENT NUMBER OF DJII'S FOUND
        MOV     R0,DJCSR(R1)    ;STORE CSR ADDRESS
        ADD     #10,R0          ;POINT TO NEXT DJII
        SOB     R2,1$           ;LOOP FOR ALL 8 POSSIBLE DJII'S
2$:     MOV     R1,DJ.NUM       ;STORE NUMBER OF DJII'S FOUND
        ASL     R1              ;*2 FOR WORD OFFSET
        MOV     R1,DJ.CSR       ;STORE OFFSET TO LAST DJII CSR
        MOV     DJCSR(R1),R0    ;GET LAST DJII CSR ADDRESS
        ADD     #4,R0           ;POINT TO LINE PARAMETER REGISTER
        MOV     R0,DJ.LPR       ;STORE LAST DJII LPR ADDRESS
        ADD     #2,R0           ;POINT TO TRANSMIT CONTROL REGISTER
        MOV     R0,DJ.TCR       ;STORE LAST DJII TCR ADDRESS
        RTS     PC

;ROUTINE TO DETERMINE .EVEN ADDRESSES

EV:     MO.     #EC.    - 1F    ;GET # OF .EVEN'S IN THE STR
```

```
                                                    ;RESET IOT VECTOR
                                                    ;SET FIRST OF ACR
                                                    ;SET CSR
                                                    ;BIT 0 = TRANS SCAN ENABLE
                                                    ;BIT 14 = TRANS INTERRUPT ENABLE
                                                    ;INC POINTER
                                                    ;ACR TO R0
                                                    ;ACR = FOR NEXT INTERRUPT
                                                    ;RESTORE IOT VECTOR

                                                    ;REPOSITION ADDRESS TO BUFFER
                                                    ;SAVE FIRST ADDRESS
                                                    ;RESET STACK FROM IOT
                                                    ;RETURN FROM INTERRUPT - RESTORE STATUS

017776  000176  ABCINT:  .WORD   @SWREG.SWR
                         .WORD   @$
                         .WORD   TMP1
                         .WORD   @$,TMP1
                         .WORD   @$,TMP1
                         .WORD   @$
                         .WORD   CNTLC
                1$:      .WORD   0

                SJSWRR:  MOV     6.-(SP)
                         MOV     4,-(SP)
                         MOV     #1$,4
                         CMP     #-1,@SWR
                         BEQ     2$
                         BR      3$
                1$:      CMP     (SP)+,(SP)+
                2$:      MOV     @SWREG,SWR
                         MOV     @DISPREG,DISPLAY
                3$:      MOV     (SP)+,4
                         MOV     (SP)+,6
                         RTS     PC

017770  022737  000176  000316  CNTLC:  CMP     @SWREG.SWR
017776  001023                          BNE     1$
000000  000004  020062                  TYPE    ,SWREG
000004  013705  000176                  MOV     SWREG,TTY       ;TYPE SWREG IN OCTA
000010  004737  016562                  JSR     PC,PRINTR       ;TYPE LEADING DESCR
000014  000004  020065                  TYPE    ,NEWIS
000020  004537  016216                  JSR     R5,READIN
000024  017706                          .WORD   TMP1
000026  001360                          BNE     CNTLC
000030  022737  000020  020050          CMP     #20.CNT
000036  001403                          BEQ     1$
000040  013777  017706  161352          MOV     TMP1,@SWR
```

```
[illegible]  000046  000127                    18.     R:S     P:
[illegible]  000050  000000                    CNT:    0
[illegible]  [illegible]  [illegible]  043516  036527  NEW1$:  .ASCIZ  "  NEW2 "
[illegible]  [illegible]  [illegible]  052522  036522  S.RED:  .ASCIZ    15  12 "S.R= "
[illegible]  [illegible]  [illegible]
[illegible]  [illegible]  000001                    .END
```

DEFADR.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

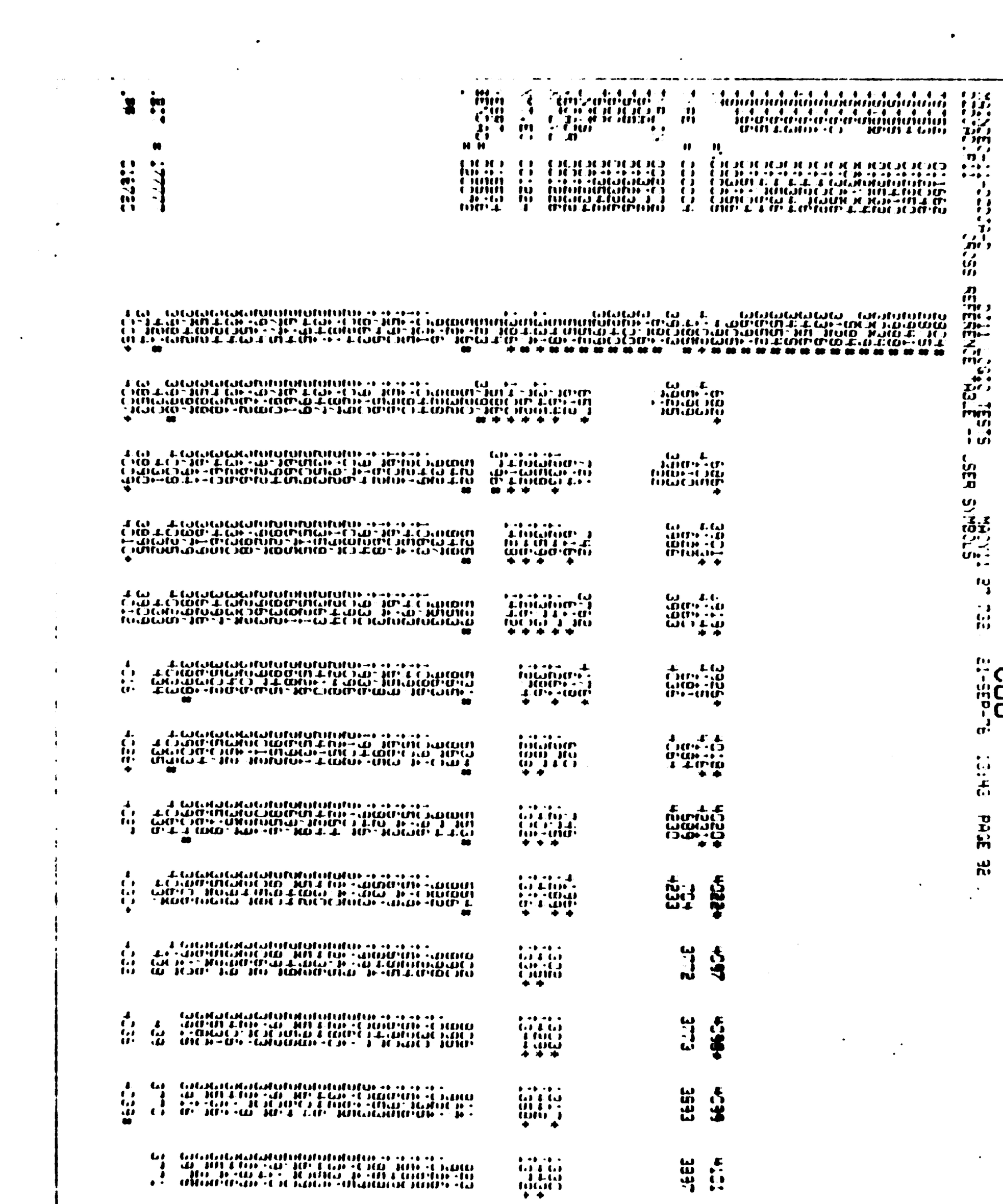
```
.ERR    017470          536    4158*
.ESS    001214          542*   3457*  3459    3463*   3464    3484*   3533*   3563*   3633*   3655*   3671
.BEGIN  001324          530     521*
.BELL   = 000007        496*   3772    3893
.TYPS   016552         4005*
.TYPO   = 000001        497*    967     969     979    1111    1133    1201    1219    1237    1255    1293    1321    1343
                       1377    1405    1433    1477    2812    2882    2946    2995    3045    3060    3398
.BIT1   = 000002        498*    800     801     810    1560    3015
.BIT10  = 002000        507*    884     885     894    2311
.BIT11  = 004000        508*   2394
.BIT12  = 010000        509*    905     906     915    2479    3244
.BIT13  = 020000        510*   2562    3140    3181    3255    3312
.BIT14  = 040000        511*    926     927     936    2645    3410
.BIT15  = 100000        512*   2728
.BIT2   = 000004        499*    821     822     831     948     962     981    1643
.BIT3   = 000010        500*    950     952    1726    2823    3315
.BIT4   = 000020        501*    956    1811    2824    2944    3373    3703    3838
.BIT5   = 000040        502*   1894
.BIT6   = 000100        503*    842     843     852    1977
.BIT7   = 000200        504*   2060
.BIT8   = 000400        505*    863     864     873    2145
.BIT9   = 001000        506*   2228
.BMEC   002336          727    7229    7328*
.CNT    020050         3940*   3968*   4237    4242*
.CRLF   012200          730    4208    4229*   4236
.CTRLC  003003          659    6603*   6615
.DBR    001210          539*   7388*   7445*   750     779     780*    781*    800*    801     805     809*    810     814*
                        822     826     830*    831     842*    843     847     851*    852     863*    864     868*
                        873     884*    885     889     893*    894     905*    906     910     914*    915     918*
                        931     935*    936     948*    950*    951     967*    968     969     973     979*    980     984*
                        994*    995*    996    1001*   1002*   1003*   1004*   1005    1019*   1020    1024*   1025    1029*
                       1030    1110*   1113    1129    1134    1149*   1154    1174    1179    1200*   1203    1220*   1223*
                       1220    1236*   1238    1249*   1264*   1266    1277*   1292*   1294    1305*   1320*   1334*   1345*
                       1348*   1350    1361*   1376*   1378    1389*   1404*   1406    1417*   1432*   1434    1454*   1465*
                       1484    1488    1527    1533*   1561    1567    1571    1610    1616*   1644    1654    1667*   1695*
                       1699*   1727    1733    1737    1776    1782*   1812    1818    1822    1861    1867*   1895*   1905*
                       1905    1944    1950*   1978    1984    1988    2027    2033*   2061    2065    2084*   2118    2135*
                       2146    2152    2156    2195    2201*   2229    2235    2239    2278    2284*   2312    2330    2355*
                       2361    2367*   2395    2401    2405    2444    2450*   2480    2486    2490    2529    2539    2546*
                       2569    2573    2612    2618*   2646    2652    2656    2695    2701*   2729    2735    2739    2746*
                       2784*   2814    2820    2823*   2824    2830    2843    2860*   2883    2886    2889    2896    2903*
                       2931*   2941*   2944    2947    2953    2966*   2967    2996    3002    3015*   3046    3051    3057*
                       3091    3094    3113*   3137    3148    3152    3180    3227    3244*   3245    3251    3253    3259*
                       3287    3302    3308    3311    3316*   3319    3340    3365*   3376*   3379    3383    3386    3392*
                       3396*   3397    3403    3418*   3436    3446*   3456*   3498    3566    3640*   3654*   3660    3667*
                       3698*   3703    3708    3715    3758*   3817*   3820*   3823*   3826*   3838    3841*   3919
.DEFADR 001312          560*    605
.DEFDEC 001314          561*
.DEFADR 001230          550*    601     603     605*    606*    607     745    4172*   4185
.DEFFLA 001320          564*   3869*   3873*   4222*
.DEFPRE 001174          526*   4222
.DELEN  001306          558*    740*    747*   1511    1594    1677    1760    1784*   1945    1989    2011    2094    2118*
                       2179    2262    2345    2428    2462*   2513    2596    2679    2762    2786*   2830*   2874    2958*

.DEMANG= 000020         515*    655
.DEPAR  001310          559*    741*    748*   3510    3596
.DEVOT  001304          557*    658*    742     744*    749*   3762
```

```
                    3763    3766*   4179
                     589    3890*
                    1245    1243    1246*
                    1268    1271    1274*
                    1296    1299    1302*
                    1325    1327    1330*
                    1353    1355    1358*
                    1381    1383    1386*
                    1409    1411    1414*
                    1437    1439    1442*
                    3051    3054    3057*
                    3095    3099    3102*
                    3730    3754*
                    5336*   5911*   3894*   3915
                    3903    3918*
                     566*    726     731*
                     595*    599*    603     608
                     689*    694     698     702
                     822*    831     833     838     840     843     856
                     902*    904*    913
                     610*    613     619     621
                     784*    793     788     803     807     812     824     828     833     845     849     854     856
                     970     975     887     891     996     908     912     917     929     933     938     954     960
                     966     971     976     983     999    1008    1022    1027    1032    1045    1050    1054    1073
                    1087    1092    1097    1117    1127    1131    1136    1158    1159    1176    1181    1205    1215
                    1222    1243    1270    1298    1324    1352    1380    1408    1436    1489    1494    1498    1508
                    1514    1520    1525    1530    1572    1577    1581    1591    1597    1603    1608    1613    1655
                    1660    1664    1674    1680    1686    1691    1696    1738    1743    1747    1757    1763    1759
                    1774    1779    1823    1828    1832    1842    1848    1854    1859    1864    1906    1911    1915
                    1925    1931    1937    1942    1947    1989    1994    1998    2008    2014    2020    2025    2030
                    2072    2077    2081    2091    2097    2103    2108    2113    2157    2162    2166    2176    2182
                    2188    2193    2198    2240    2245    2249    2259    2265    2271    2276    2281    2323    2329
                    2332    2342    2349    2354    2359    2364    2406    2411    2415    2425    2431    2437    2442
                    2447    2491    2496    2500    2510    2516    2522    2527    2532    2574    2578    2583    2593
                    2599    2605    2610    2615    2657    2662    2666    2676    2682    2698    2693    2698    2740
                    2745    2749    2759    2765    2771    2776    2791    2829    2833    2846    2853    2857    2891
                    2903    2907    2911    2915    2919    2923    2928    2956    2964    2972    2977    3005    3013
                    3022    3053    3059    3063    3096    3104    3108    3142    3154    3163    3167    3195    3199
                    3203    3207    3218    3230    3248    3260    3268    3272    3279    3290    3305    3314    3322
                    3327    3331    3338    3347    3353    3358    3400    3407    3412    3417    3425    3430    3439
                    3461    3466    3493    3498    3503    3508    3514    3519    3524    3528    3532    3537    3547
                    3577    3593    3589    3600    3606    3612    3620    3627    3632    3665    3669    3673    3677
                    3681    3718    3723    3737    3743    3836    4083    4178
HLTADR  016144      3898*   3899*   3900    3901    3916*
HLTCNT  016142      3900*   3915*   3922*
ICNT    001200       535*    571     725     765*   3856    3862    3864   3866*   3867*   3869   3872*   3873   3879
                    3912*
ILLUP   017072      4040    4067*
INHRE   016510      3933*   3935*   3989    3997*
INITA   015536      3478    3554    3817*
INITB   015546      3038    3083    3820*
INITC   015556      2988    3823*
INITD   015566      1470    1553    1636    1719    1804    1987    1970    2053    2138    2221    2304    2387    2472
                    2555    2639    2721    2804    2875    3128    3826*
INITR   015574      3818    3821    3824    3828*
INPUT   016512       625     662     670     672     674     691     693     695*    699*    706     709     713     713
```

```
                        3941    3971    3972    3998*
ICTRAP  017104           576    4077*
.OT$    017130          4078    4097*
.SR22   004464          1232    1242*
.SR23   004644          1260    1270*
.SR24   004664          1288    1298*
.SR25   005006          1316    1327*
.SR26   005126          1344    1355*
.SR27   005246          1372    1383*
.SR30   005366          1400    1411*
.SR31   005506          1428    1439*
.SR56   012620          3034    3053*
.SR57   012774          3079    3099*
.SR66   015274          3705    3735*
KBDINT  017640          3763    3853    3890    3909    4201*
KIT$    015742          3859    3965    3972*   3913
LAD     015770           766*   1101*   1451*   1785*   2119*   2453*   2794*   3868*   3874    3876    3879*
LENGTH  001236           553*    665     678     747
LEVEL?= 000340           513*    778
LOP32   005574          1478*   1479
LOP33   005774          1561*   1562
LOP34   006174          1644*   1645
LOP35   006374          1727*   1728
LOP36   006606          1812*   1813
LOP37   007006          1895*   1896
LOP40   007206          1978*   1979
LOP41   007406          2061*   2062
LOP42   007620          2146*   2147
LOP43   010020          2229*   2230
LOP44   010220          2312*   2313
LOP45   010420          2395*   2396
LOP46   010632          2480*   2481
LOP47   011032          2563*   2564
LOP50   011232          2646*   2647
LOP51   011432          2729*   2730
LOP61   014060          3383*   3384    3444
LOP64   014550          3562*   3638
MSGADR  017244           599    4122*
MSGCON  017343           660    4134*
MSGDAS  017410           684    4141*
MSGLEN  017414           689    4142*
MSGLIN  017377           681    4139*
MSGNUM  017320           622    4130*
MSGPAR  017435           704    4145*
MSGVEC  017274           610    4126*
N       = 000067         421*    770     794*    815*    836*    857*    878*    899*    920*    941*    998*   1013*   1035*
                        1053*   1077*   1100*   1102    1141*   1192*   1226*   1254*   1283*   1310*   1339*   1366*   1394*
                        1422*   1450*   1452    1535*   1618*   1701*   1784*   1786    1869*   1952*   2035*   2118*   2131
                        2203*   2286*   2369*   2452*   2454    2537*   2620*   2703*   2796*   2795    2867*   2934*   2983*
                        3028*   3073*   3118*   3370*   3450*   3469*   3545*   3643*   3644    3695*   3761*
NEXIS   020052          4233    4244*
OPEN    = 000000         514*
OVER$   015746          3857    3873*
PARITY  001276           554*    679     718     3510    3596
PC      =000307          495*    572*    596*    623*    661*    683*    687*    690*    705*    730*   1472*   1553*   1636*
                        1719*   1804*   1887*   1970*   2053*   2138*   2221*   2304*   2387*   2472*   2555*   2638*   2721*
```

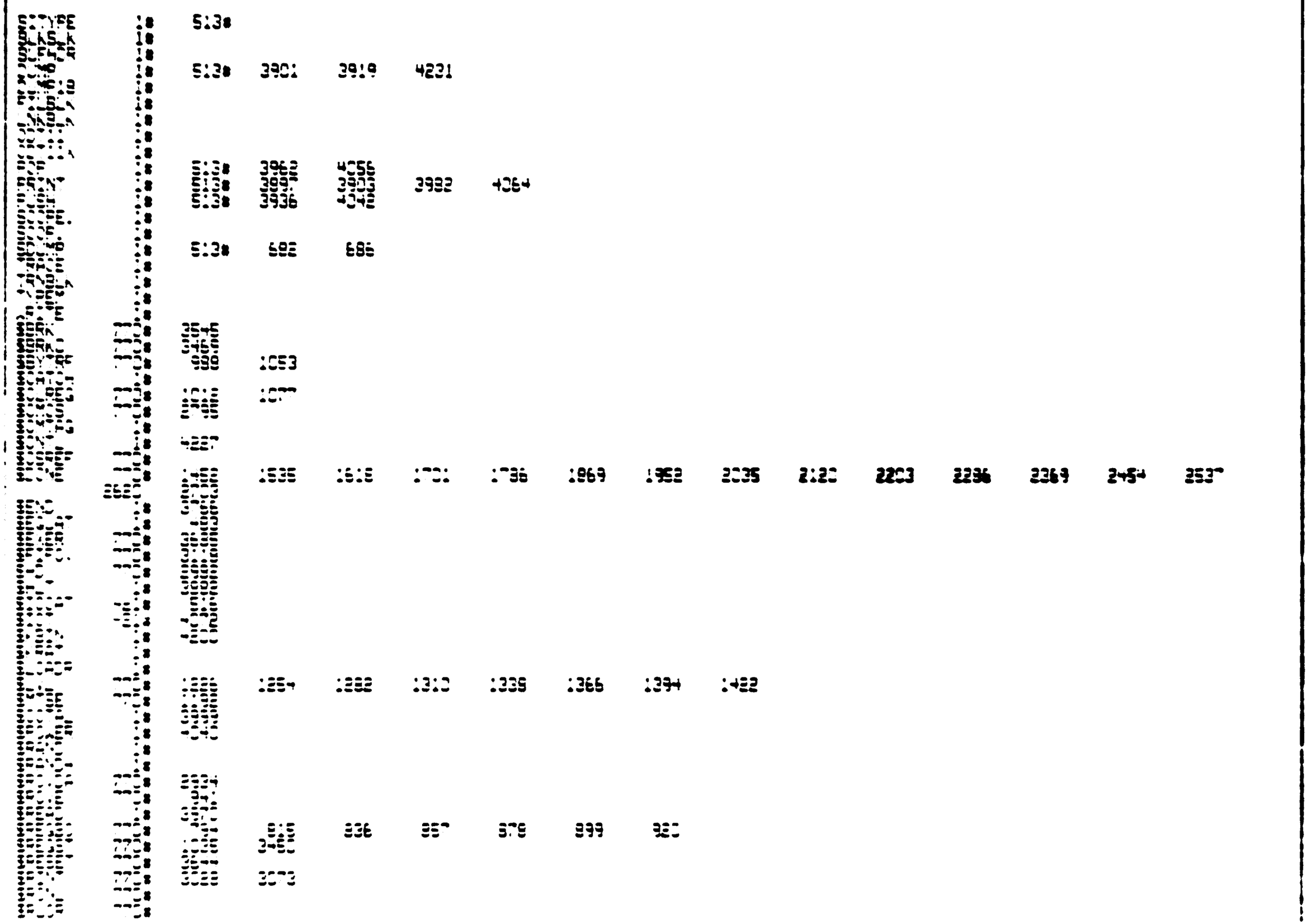
```
                    2824*  2875*  2888*  3038*  3083*  3128*  3478*  3554*  3662   3767*  3842*  3853*  3932*
                    3902*  3904*  3909*  3920*  3929*  3934*  3995*  4038*  4193*  4208*  4209*  4235*  4232*
                    4240*
FONT    001204       537*   593*   593*  3768*  3769*  3794
FDOWN$  016744        57$   4040*  4062
PRINTR  016562      3902   3920   3927   4007*  4232
PRINTS  016572       693    687   4009*
PS    = 177776       486*  1234*  1235*  1250*  1262*  1263*  1278*  1290*  1291*  1306*  1318*  1319*  1334*
                    1346*  1347*  1362*  1374*  1375*  1390*  1402*  1403*  1418*  1430*  1421*  1446*  3036*
                    3037*  3069*  3081*  3082*  3114*  3240*  3241*  3299*  3300*  3366*  3652*  3694*  3695*
                    3961   3965*
FSTEMP  016362      3961*  3965   3968*
PUP$    017010      4049   4052*
PUVEC$  017100      4040*  4041*  4049*  4071*
RBUF    001212       540*   752*  1492   1523   1575   1606   1658   1689   1741   1772   1826   1857   1909
                    1940   1992   2023   2075   2106   2160   2191   2243   2274   2326   2357   2409   2440
                    2494   2525   2577   2608   2660   2691   2743   2774   2827   2855   2905   2913   2921
                    2962   2975   3011   3020   3057   3102   3201   3261   3256   3325   3356   3405   3428
                    3501   3535   3581   3630   3675   3721   3735   3750
RCVLVL  001222       546*   760*  3035*  3065   3066*  3080*  3110   3111*  3243*  3360   3361*  3706*  3755
                    3756*
RCVVEC  001220       545*   739*   746*   758   3034*  3065*  3079*  3110*  3242*  3301*  3360*  3705*  3755*
READIN  016216       600    611   3933*  4234
READ$   016364       623    661    690    705   3934   3970*
RESTAR  002302       532    597    663    671    673    724*  3764   3792   4065
RETURN  017236      4120*
R0    =000000       487*   573*   574*   575*   576*   577*   578*   579*   580*   581*   582*   583*   585*
                    664*   668*   676*   720*   721    732*   733    734*   735*   736   1112*  1115*  1121
                   1153*  1156*  1164   1202*  1207*  1211*  1214*  1481*  1486*  1491   1564*  1569*  1574
                   1647*  1652*  1657   1730*  1735*  1740   1815*  1820*  1825   1898*  1903*  1909   1981*
                   1986*  1991   2064*  2069*  2074   2149*  2154*  2159   2232*  2237*  2242   2315*  2320*
                   2325   2398*  2403*  2408   2483*  2488*  2493   2566*  2571*  2576   2649*  2654*  2659
                   2732*  2737*  2742   2788*  2789*  2790*  2791*  2793   2839*  2848*  2893*  2898*  2950*
                   2959*  2970*  2999*  3008*  3136*  3144*  3147*  3156*  3177*  3183*  3200*  3221*  3265*
                   3281*  3324*  3341*  3343   3389*  3394*  3485*  3491*  3562*  3569*  3696*  3707   3731*
                   3784*  3828*  3830*  3833*  3834*  4042   4061*  4159*  4163   4165*  4171*  4172
R1    =000001       488*   626*   628    630    632*   645    646*   647*   648*   650*   651*   653*   655
                    657    665*   666*   677*   682    685*   686    688*   716    750*   751*   752    753*
                    754    756*   757    758*   759    760    761    762    763    764    787*   951*   952
                    956    962    968*   980*   996*   997   1005*  1006   1020*  1030*  1042*  1043   1048*
                   1061*  1062   1070*  1071   1085*  1095*  1113*  1122*  1123*  1124*  1125   1134*  1150*
                   1167   1186*  1203*  1212*  1220*  1238*  1266*  1294*  1322*  1350*  1378*  1406*  1434*
                   1478*  1488*  1492*  1496   1503   1512   1519   1523*  1527*  1528   1561*  1571*  1575*
                   1579   1586   1595   1601   1606*  1610*  1611   1644*  1654*  1659*  1662   1669   1679
                   1684   1689*  1693*  1694   1727*  1737*  1741*  1745   1752   1761   1767   1772*  1776*
                   1777   1812*  1822*  1826*  1830   1837   1846   1852   1857*  1861*  1862   1935*  1935*
                   1909*  1913   1920   1929   1935   1940*  1944*  1945   1978*  1988*  1992*  1996   2003
                   2012   2018   2023*  2027*  2028   2061*  2071*  2075*  2079   2086   2095   2101   2106*
                   2110*  2111   2145*  2156*  2160*  2164   2171   2180   2196   2191*  2195*  2196   2239*
                   2239*  2243*  2247   2254   2263   2269   2274*  2278*  2279   2312*  2322*  2326*  2330
                   2337   2346   2352   2357*  2361*  2362   2395*  2405*  2409*  2413   2420   2429   2435
                   2440*  2444*  2445   2480*  2490*  2494*  2498   2505   2514   2520   2525*  2529*  2530
                   2563*  2573*  2577*  2581   2588   2597   2603   2608*  2612*  2613   2646*  2656*  2660*
                   2664   2671   2680   2686   2691*  2695*  2696   2729*  2739*  2743*  2747   2754   2763
                   2769   2774*  2778*  2779   2814*  2827*  2830*  2831   2843*  2944   2951   2955*  2993*
                   2886*  2889*  2896*  2901   2905*  2909   2913*  2917   2921*  2925*  2926   2947*  2953*
```

```
                    2954    2962*   2967*   2968    2975*   2996*   3002*   3003    3011*   3020*   3046*   3057*   3061
                    3091*   3102*   3106    3137*   3140    3148*   3152*   3161    3165    3180*   3181    3187    3201*
                    3205    3213    3216    3227*   3228    3245*   3246    3254*   3255    3259*   3266*   3270    3274
                    3277    3287*   3288    3302*   3303    3311*   3312    3319*   3320    3325*   3329    3333    3336
                    3340*   3345    3351    3356*   3381*   3423    3442*   3458*   3459    3464*   3488*   3489    3496
                    3501*   3506    3512    3517    3522    3526    3530    3535*   3560*   3593    3618    3636*   3662*
                    3667*   3671*   3675*   3679*   3691*   3727*   3728    3936    3939*   3944*   3951*   3954*   3955*
                    3957*   3964*   4043    4053*   4054*   4060*   4080*   4161*   4166*   4176    4185*   4186*   4189
                    4190*
R2      =%000002     489*    625*    626     635     637     639     641     678*    696*    697*    701     703*    1129*
                    1154*   1165*   1166*   1167    1179*   1503*   1504*   1505*   1506    1511*   1512    1517*   1518
                    1586*   1587*   1588*   1589    1594*   1595    1600*   1601    1669*   1670*   1671*   1672    1677*
                    1678    1683*   1684    1752*   1753*   1754*   1755    1760*   1761    1766*   1767    1937*   1938*
                    1939*   1940    1945*   1946    1951*   1952    1920*   1921*   1922*   1923    1928*   1929    1934*
                    1935    2003*   2004*   2005*   2006    2011*   2012    2017*   2018    2086*   2087*   2088*   2089
                    2094*   2095    2100*   2101    2171*   2172*   2173*   2174    2179*   2180    2185*   2186    2254*
                    2255*   2256*   2257    2262*   2263    2268*   2269    2337*   2338*   2339*   2340    2345*   2346
                    2351*   2352    2420*   2421*   2422*   2423    2428*   2429    2434*   2435    2505*   2506*   2507*
                    2508    2513*   2514    2519*   2520    2588*   2589*   2590*   2591    2596*   2597    2602*   2603
                    2671*   2672*   2673*   2674    2679*   2680    2685*   2686    2754*   2755*   2756*   2757    2762*
                    2763    2768*   2769    3213*   3214*   3215*   3216    3274*   3275*   3276*   3277    3333*   3334*
                    3335*   3336    3383*   3386*   3392*   3397*   3398    3405*   3410    3415    3420    3428*   3436*
                    3437    3566*   3567    3575    3581*   3597    3598    3604    3610    3615    3625    3630*   3715*
                    3716    3721*   3735*   3741    3750*   3937    3941*   3942    3945    3963*   4044    4059*   4082*
                    4159*   4169*
R3      =%000003     490*    641*    642*    651     679*    714*    715*    718*    1174*   3420*   3421*   3422*   3423
                    3593*   3594*   3595*   3615*   3616*   3617*   3618    3692*   3713    3726*   3741    3747*   3748*
                    3938    3945*   3946    3948*   3949    3957    3962*   3970    3971*   3972    3994*   4045    4058*
R4      =%000004     491*    680*    714     719*    1151*   1152    1178    1187*   2813*   2816    2817*   3561*   3563
                    3637*   3693*   3710    3711*   4011    4012*   4032    4037*   4046    4057*
R5      =%000005     492*    600*    611*    949*    958*    1482*   1565*   1648*   1731*   1816*   1893*   1982*   2065*
                    2150*   2233*   2316*   2399*   2484*   2567*   2650*   2733*   2840*   2994*   2951*   3000*   3150*
                    3178*   3223*   3283*   3390*   3486*   3540*   3564*   3939    3966*   4047    4056*   4234*
SCOPE = 104400       483*    767     793     814     835     856     877     898     919     940     987     1011    1034
                    1052    1076    1099    1139    1190    1224    1252    1280    1308    1336    1364    1392    1420
                    1448    1534    1617    1700    1783    1868    1951    2034    2117    2200    2285    2368    2451
                    2536    2619    2702    2785    2862    2932    2981    3026    3071    3116    3368    3448    3469
                    3543    3641    3683    3759
SP      =%000006     494*    571*    645*    649*    650     725*    787     790     1243*   1271*   1299*   1327*   1355*
                    1383*   1411*   1439*   3054*   3099*   3262*   3362*   3933    4048    4052*   4077    4079*   4080
                    4081    4082    4183*   4192    4195*   4196    4197    4214*   4215*   4220    4223    4224
SPACE   017241       3925    4121*
SUSWRR  017710        572    4214*
SVLAD$  015722       3861    3867*   3875
SWR     001316        563*    728    3770    3775    3854    3856    3858    3860    3991    3995    3935    3909    4201
                    4217    4221*   4228    4239*
SWREG   000176        527*    728    4201    4221    4228    4231
SWREQ   020062       4230    4246*
SW10    = 002000      435*   3770    3890    3891
SW11    = 004000      434*   3860
SW12    = 010000      433*   3771    3774    3775
SW13    = 020000      432*   3895
SW14    = 040000      431*   3858
SW15    = 100000      430*
SW8     = 000400      438*   3853    3854
SW9     = 001000      437*   3909
```

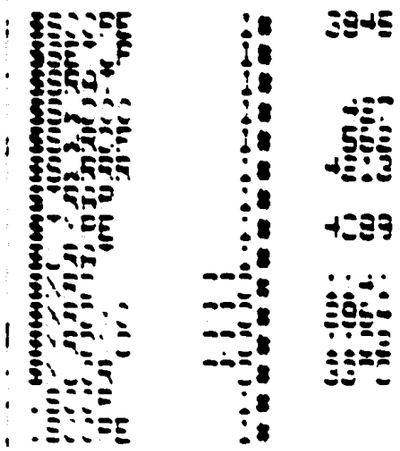


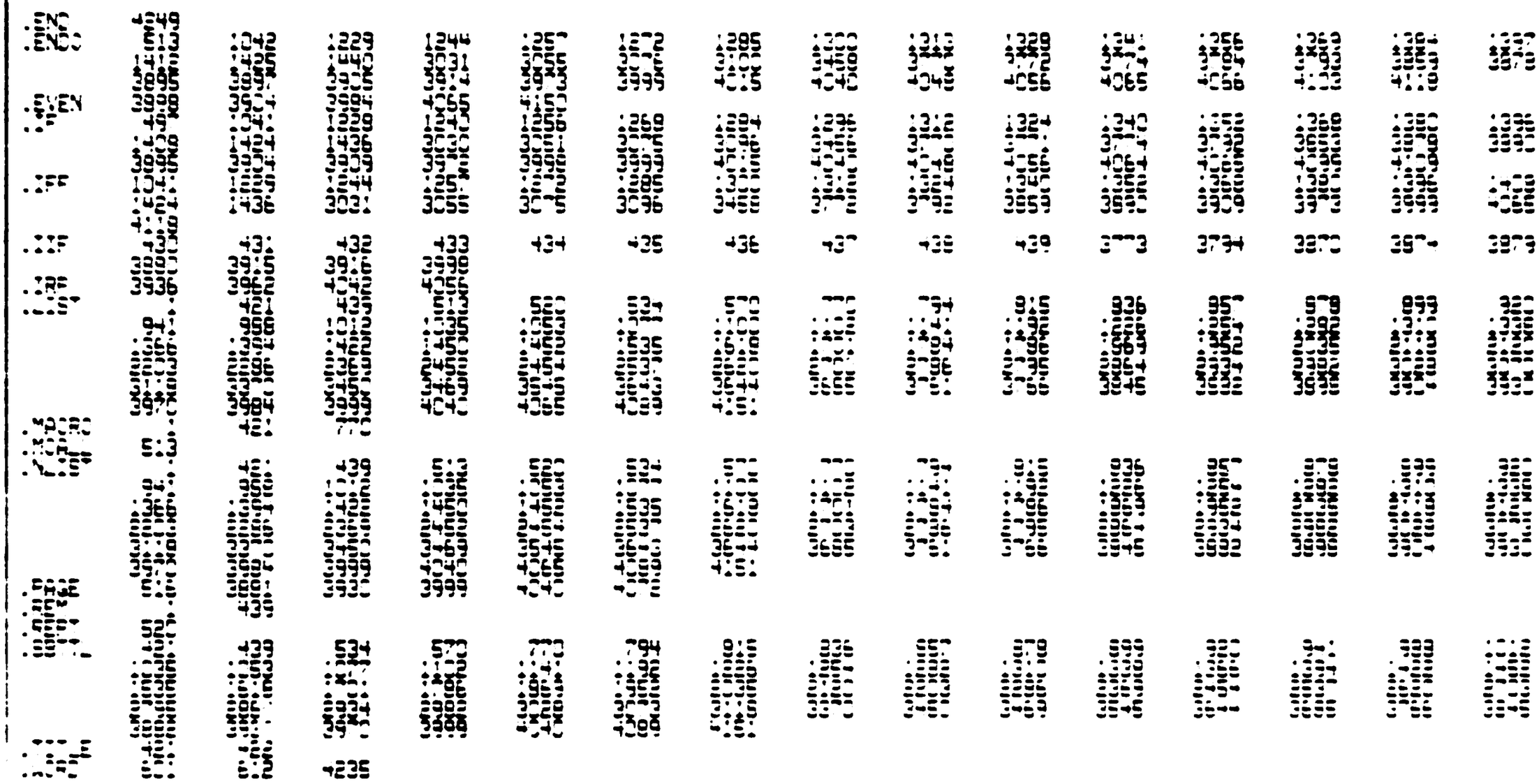
[illegible].P11 CROSS REFERENCE TABLE -- USER SYMBOLS

.ERF	016640	4014	4017	4022#		
.DS	[illegible]	4158	4231			
[illegible]	[illegible]	[illegible]	[illegible]			
[illegible]	[illegible]	[illegible]	[illegible]	4073#		
[illegible]	[illegible]	[illegible]	[illegible]			
[illegible]	[illegible]	3985#	3986	4101#	4102	4118#



CROSS REFERENCE TABLE -- MACRO NAMES





K08

Spooler runtime 16 Seconds, 71 KCS, 416 disk reads, 3 disk writes, 97 pages