DJ11

LOGIC TEST

MD-11-DZDJA-D

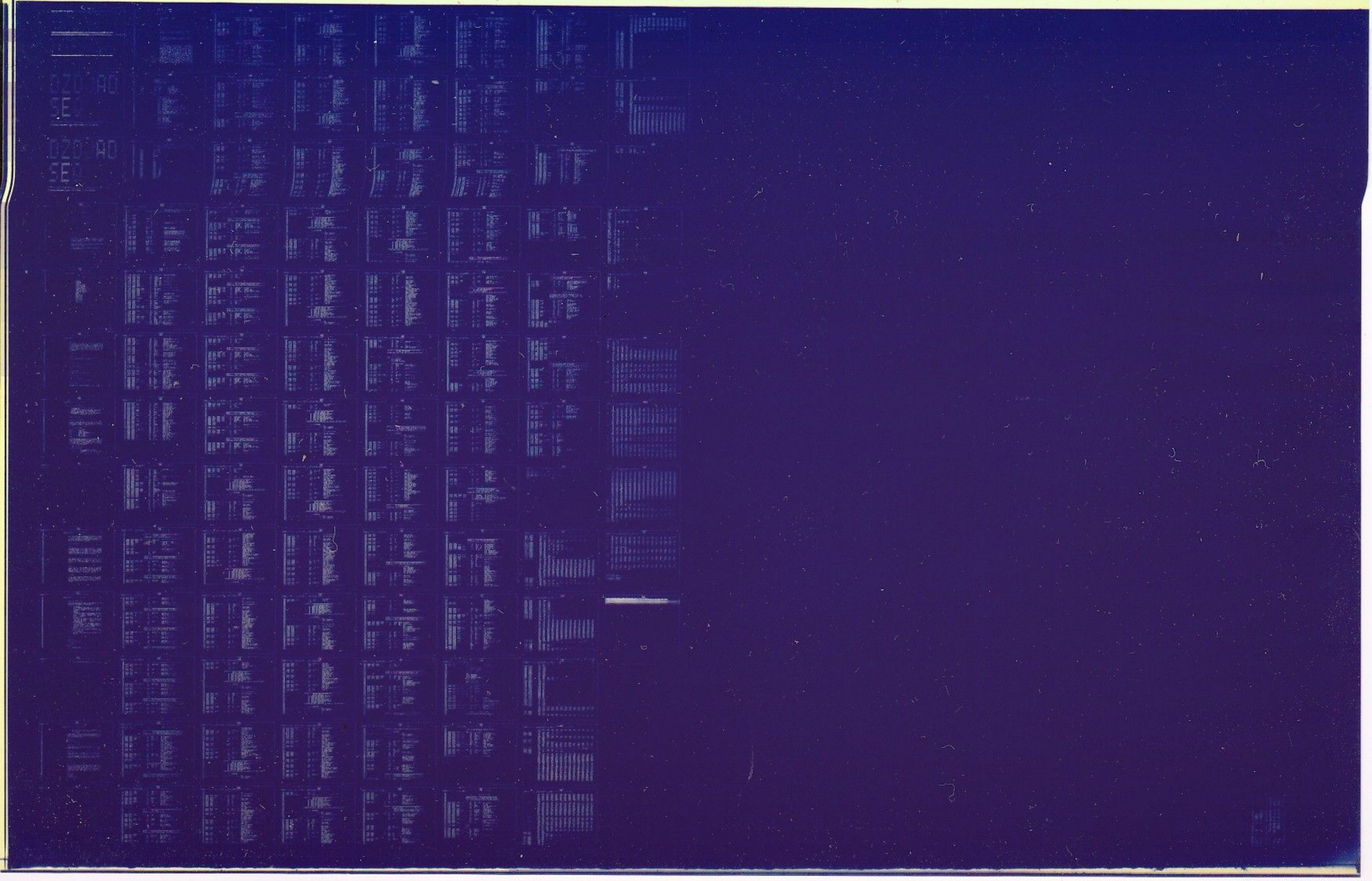
EP-DZDJA-D-DL-A

FICHE 1 OF 1

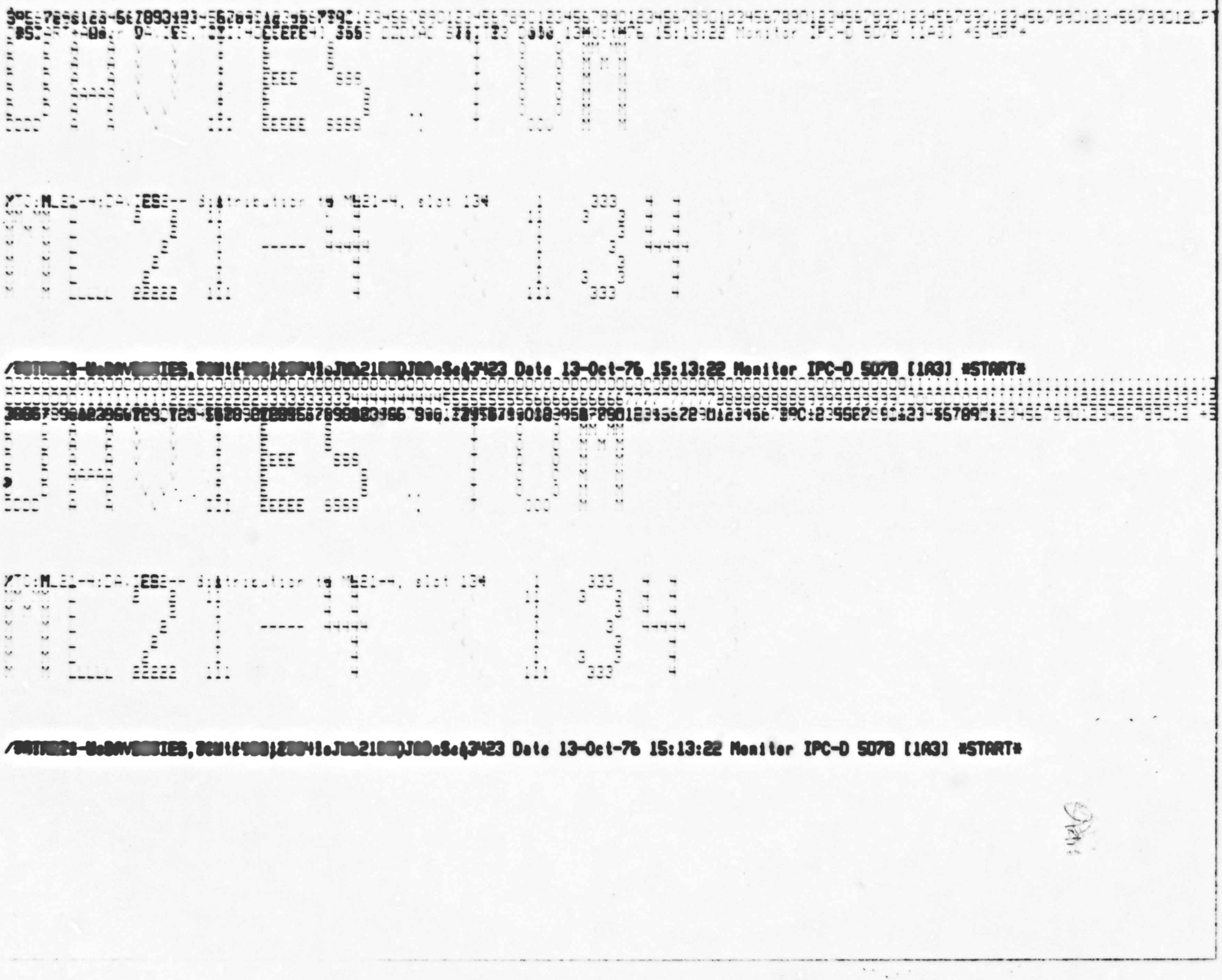
NOV 1976

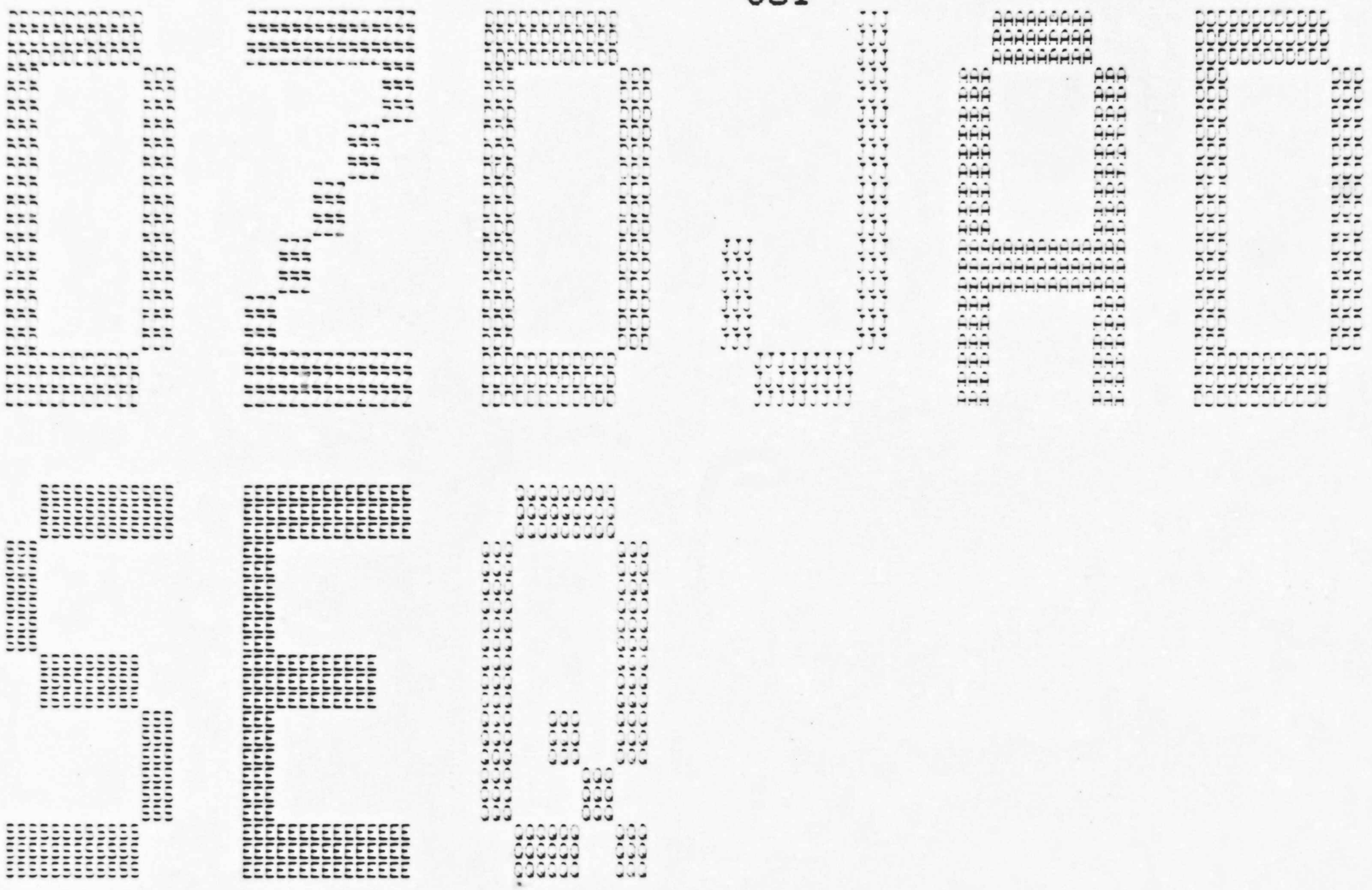
digital

MADE IN USA



B01

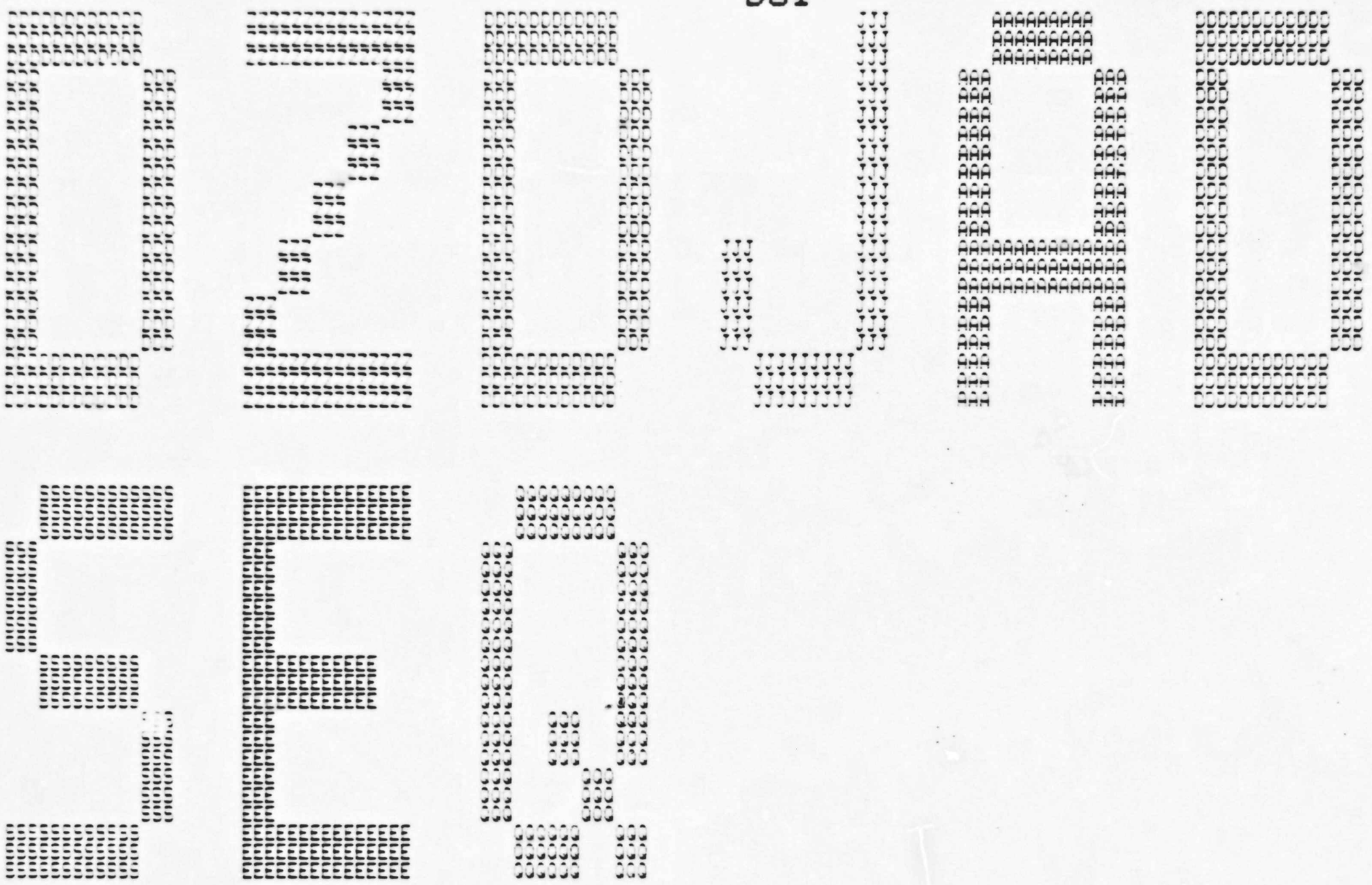





```
-PTSR- Version 6(1003+7), Running on MTA1
*S*AB-* User DAVIES,TOM [400,27041] Job  DZDJAD Seq. 83 Date 13-Oct-76 15:13:22 Monitor IPC-D SCTS [1A3] *START*
Request created: 13-Oct-76  14:51:19     /TO:ML21-4:DAVIES -- distribution to ML21-4, slot 134

File will be RENAMED to "DSK" protection
```

D01



```
LPTSPL  Version 6(100344)  Running on MTA1
 *START* User DAVIES,TOM [400,2704] Job  DZDJAD Seq. 23 Date 13-Oct-76 15:13:22 Monitor IPC-D 5078 [1A3] *START*
Request created: 13-Oct-76  14:51:19     /TO:ML21-4:DAVIES -- distribution to ML21-4, slot 134

                                                                                                    15:13:27
File will be RENAMEd to (057) protection
```

.REM %

IDENTIFICATION

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

CONTENTS

1. ABSTRACT

THIS PROGRAM TESTS THE LOGIC OF THE DJ11 ASYNCRONOUS MULTIPLEXER IN MAINTENANCE MODE. IT CHECKS THAT ALL THE CONTROL REGISTERS FUNCTION PROPERLY, THAT INTERRUPTS OCCURE AT THE RIGHT LEVEL, AND THAT DATA CAN BE TRANSMITTED AND RECEIVED CORRECTLY. THIS PROGRAM DOES NOT TEST THAT THE INPUT AND OUTPUT LEAD CONNECTIONS ARE FUNCTIONAL. (SEE MAINDEC-11-DZDJB, PROGRAMS 2 AND 3 FOR ON-LINE TESTING). THE PROGRAM SHOULD BE RUN FOR AT LEAST 2 PASSES WITH ALL SWITCHES DOWN.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD COMPUTER WITH CONSOLE TELETYPE
UP TO 16 DJ11 ASYNCRONOUS MULTIPLEXERS.

2.2 STORAGE

THIS PROGRAM USES ALL OF 8K, EXCEPT ABS LOADER.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200. IT MAY BE RESTARTED AT 1000 AFTER ALL PARAMETERS HAVE BEEN SELECTED.

4.3 PROGRAM AND OPERATOR ACTION

1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
2) LOAD ADDRESS 200.
3) IF HARDWARE SWITCH REGISTER IS AVAILABLE, SET SWITCHES (SEE SEC. 5.1), ALL DOWN FOR WORST CASE, PRESS START.
4) IF SWITCH-LESS PROCESSOR SIMPLY PRESS START.
5) ENTER PARAMETERS (SEE SEC. 5.3) AS THEY ARE REQUESTED.
6) THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS.
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 T-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

```
  198                                   COMMITTED KEYING IN SWREG VALUE.
```

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

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL (VIA A TRAP INSTRUCTION) IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF "LAD" MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE (CALLED BY AN EMT INSTRUCTION) PRINTS OUT AN ERROR MESSAGE (SEE 6.1). IF SW<9> IS ON A 1 AND A HLT IS EXECUTED, THE SUBTEST WILL BE LOOPED UPON UNTIL 16 CONSECUTIVE GOOD PASSES ARE COMPLETED. TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1. TO RING THE BELL ON AN ERROR, PUT SW<10> ON A 1.

5.2.4 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 1 "P(REVIOUS)" SELECTS THE NUMBER OF UNITS PREVIOUSLY SELECTED.

4) "STANDARD CONFIGURATION? "
"Y(ES)" OR DEFAULT (CARRIAGE RETURN) SELECTS 8 LEVEL CODE, NO PARITY. "N(O)" CAUSES REQUESTS FOR CODE LEVEL AND PARITY ON ALL REQUESTED LINES IN GROUPS OF FOUR. "P(REVIOUS)" SELECTS THE CODE LEVELS AND PARITIES PREVIOUSLY SELECTED.

5) "CHAR LENGTH: "
THE CODE LEVEL FOR THE LINE GROUP SPECIFIED. MUST BE 5, 6, 7, OR 8. THE DEFAULT (CARRIAGE RETURN) IS TO 8 LEVEL CODE.

6) "PARITY (NO, ODD, EVEN): "
THE TYPE OF PARITY SELECTED FOR THE LINE GROUP SPECIFIED. THE DEFAULT (CARRIAGE RETURN) IS TO NO PARITY.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

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

WHERE:

ADR = ADDRESS OF ERROR HLT
DJADR = CSR ADDRESS OF DJ11 UNDER TEST

```
305                                     (RN)    = CONTENTS OF GENERAL REGISTER "N",  FROM  NONE  TO
306                                               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
```

```
400 00:03:15
1000 00:01:45
1800 00:01:20
2400 00:00:55
3000 00:00:30
3600 00:00:15
```

8.2 STACK POINTER

STACK IS INITIALLY SET TO 1800

8.3 PASS COUNT

A 32 BIT (2 WORDS) PASS COUNT IS KEPT IN "PASSES" (1804,1806). IT CAN BE CLEARED FROM THE CONSOLE, BY RESTARTING AT 200, OR BY RELOADING THE PROGRAM.

8.4 POWER FAIL

EACH TEST CAN BE POWER FAILED WITH NO ERRORS. TO USE, START THE TEST AS USUAL AND POWER DOWN THEN UP AT ANY TIME. THE ROUTINE SHOULD TYPE "POWER" AND RESTART THE PROGRAM WITH NO OTHER ERROR TYPEOUTS.

9. PROGRAM DESCRIPTION

THIS PROGRAM TESTS THE LOGIC OF UP TO 16 DJ11 ASYNCHRONOUS DATA MULTIPLEXERS IN MAINTENANCE MODE. IT CHECKS THAT ALL THE CONTROL REGISTERS FUNCTION PROPERLY, THAT INTERRUPTS OCCUR AT THE RIGHT PRIORITY LEVEL, AND THAT DATA CAN BE TRANSMITTED AND RECEIVED CORRECTLY. THE PROGRAM HAS MANY SUBTESTS (THE CODE BETWEEN 2 SCOPE STATEMENTS) WHICH ARE RUN 16 TIMES BEFORE CONTINUING TO THE NEXT. SW(11) ON A CAUSES EACH SUBTEST TO BE RUN ONLY ONCE. SW(9) ON A ENABLES LOOP ON ERROR. THE ADDRESS ICNT (LOC 1800) CONTAINS THE ITERATION COUNT IN THE LEFT BYTE AND THE TEST NUMBER IN THE RIGHT BYTE. ALL THE SUBTESTS SHOULD BE RUN SEQUENTIALLY BY STARTING AT 200 NOT BY STARTING AT THE BEGINNING OF THE SUBTEST. TO LOOP ON A PARTICULAR SUBTEST, PUT THE TEST NUMBER (SEE LISTING) IN THE RIGHT BYTE OF THE SW REGISTER AND SW(9) ON A 1. THIS TEST WILL BE LOOPED UPON UNTIL SW(8) IS PUT ON A 0 OR THE RIGHT BYTE IS CHANGED. IF THE TEST IS NON-EXISTANT, THE PROGRAM WILL BE RUN AS USUAL. IF MORE THAN ONE DJ11 IS SELECTED, ALL THE SUBTESTS ARE PERFORMED ON ONE UNIT AT A TIME. THE BELL WILL NOT RING UNTIL A PASS HAS BEEN MADE ON EACH OF THE DJ11'S SELECTED FOR TEST.

```
                        .TITLE MAINDEC-11-D2DJA-D       DJ11 LOGIC TESTS
                        .ENABLE ABS
                        .ENABLE AMA
                        ;COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
                        ;PROGRAM BY KEN CHAPMAN

                        ;     SWITCH                    USE
                        ;     ------                    ---------------------
                        ;SW15=   100000          ;HALT ON ERROR
                        ;SW14=   40000           ;LOOP ON TEST
                        ;SW13=   20000           ;INHIBIT ERROR TYPEOUTS
                        ;SW12=   10000           ;INHIBIT TRACE TRAP
                        ;SW11=   4000            ;INHIBIT ITERATIONS
                        ;SW10=   2000            ;0 - BELL ON PASS COMPLETE
                        ;                        ;1 - BELL ON ERROR
                        ;SW9=    1000            ;LOOP ON ERROR
                        ;SW8=    400             ;LOOP ON TEST IN SW(7:0)
                        .REM!

                        DJ11 REGISTER BIT ASSIGNMENTS:

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

                        RECEIVER BUFFER REGISTER (RBUF) XXXXX2 (READ ONLY)

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

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

                                BIT0-15 STOP THE SCANNER ON CORRESPONDING LINE

                        TRANSMITTER BUFFER (TBUF) XXXXX6

                                BIT0-7  TRANSMITTED CHARACTER (WRITE ONLY)
```

```
                                          BIT8-11 LINE NUMBER (READ ONLY)

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

                                          BIT0-15 TRNSMIT A BREAK ON CORRESPONDING LINE!

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

```
                    000000          . =     0                       ;TRAP CATCHER IN LOCATIONS 0 THRU 776
000000  000000  000000              0,0                             ;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


                    000046          . =     46
000046  015516                      $ENDAD

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

                    000200          . =     200
000200  000137  001324              JMP     BEGIN                   ;200 ALWAYS IS THE STARTING ADDRESS
                    001000          . =     1000                    ;RESTART ADDRESS
001000  000137  002302              JMP     RESTAR

                    001200          . =     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

001210  160010                      CSR:    160010                  ;CONTROL STATUS REGISTER(DJ UNDER TEST)
001212  160012                      RBUF:   160012                  ;RECEIVER BUFFER REGISTER(DJ UNDER TEST)
001214                              TCR:                            ;TRANSMITTER CONTROL REGISTER(DJ UNDER TEST)
001214  160014                      BCSR:   160014                  ;BREAK STATUS REGISTER(DJ UNDER TEST)
001216  160016                      TBUF:   160016                  ;TRANSMITTER BUFFER REGISTER(DJ UNDER TEST)

001220  000300                      RCVVEC: 300                     ;RECEIVER INTERUPT VECTOR ADDRESS(DJ UNDER TEST)
001222  000302                      RCVLVL: 302
001224  000304                      XMTVEC: 304                     ;TRANSMITTER INTERUPT VECTOR ADDRESS(DJ UNDER TEST)
001226  000306                      XMTLVL: 306

001230  160010                      DEVADR: 160010                  ;FIRST DEVICE ADDRESS(SELECTED)
001232  000300                      VECADR: 300                     ;FIRST VECTOR ADDRESS(SELECTED)
001234  000001                      UNITS:  1                       ;NUMBER OF UNITS TO BE TESTED
001236  000040                      LENGTH: .BLKB   40              ;TABLE OF CHARACTER LENGTHS (MASKS)
001276  000004                      PARITY: .BLKB   4               ;TABLE OF ODD PARITY FLAGS
001302  000000                      TIMER:  0                       ;USED TO SAVE RELATIVE TIMES

001304  000000                      DJUUT:  0                       ;UNIT NUMBER OF DJ11 UNDER TEST
001306  000000                      DJLEN:  0                       ;CHAR MASK TABLE POINTER
001310  000000                      DJPAR:  0                       ;UNIT FLAG (FOR ODD PARITY FLAG)
001312  160010                      DEFADR: 160010                  ;DEFAULT FIRST DEVICE ADDRESS
001314  000300                      DEFVEC: 300                     ;DEFAULT FIRST VECTOR ADDRESS

001316  177570                      SWR:    177570
001320  177570                      DISPLAY:177570


001322  177777                      FTIME:  -1
```

```
001324  012706  001200          BEGIN:  MOV     #ICNT,  SP      ;SET UP STACK POINTER
001330  004737  017710                  JSR     PC,     SUSWRR  ;CHECK FOR SWITCH REGISTER
001334  012700  000014                  MOV     #14,R0
001340  012720  015534                  MOV     #YESRT, (R0)+   ;TRACE TRAP VECTOR (14)
001344  012720  000340                  MOV     #340,(R0)+
001350  012720  017104                  MOV     #IOTRAP,(R0)+   ;IOT VECTOR (20)
001354  012720  000340                  MOV     #340,   (R0)+
001360  012720  016744                  MOV     #PDOWN$,(R0)+   ;POWER FAIL VECTOR (24)
001364  012720  000340                  MOV     #340,   (R0)+
001370  012720  015774                  MOV     #EMT$,  (R0)+   ;EMT VECTOR (30)
001374  012720  000340                  MOV     #340,   (R0)+
001400  012720  015632                  MOV     #TRAP$, (R0)+   ;TRAP VECTOR (34)
001404  012720  000340                  MOV     #340,   (R0)+
001410  012737  001446  000010          MOV     #1$,@#10
001416  006700                          SXT     R0              ;CHECK FOR PDP-11/40 OR 45
001420  012737  000006  015534          MOV     #RTT,@#YESRT
001426  012737  000006  000006          MOV     #RTT,@#6
001434  012737  000400  177774          MOV     #400,  @#177774 ;SET UP STACK LIMIT TO 1000
001442  005037  000006                  CLR     @#6
001446  012737  000012  000010  1$:     MOV     #12,@#10
001454  005037  001202                  CLR     ERRORS          ;CLEAR ERROR COUNTER
001460  005037  001204                  CLR     PCNT            ;CLEAR PASS COUNTER
001464  005037  001206                  CLR     PCNT+2
001470  005737  000042                  TST     @#42            ;CHECK FOR ACT11 OR DDP PRESENT
001474  001404                          BEQ     GETADR          ;BRANCH IF NONE
001476  004737  017470                  JSR     PC,     AUTO    ;GO TO SUBROUTINE TO "MAP" DJ11 ON THE SYS
001502  000137  002302                  JMP     RESTAR          ;SKIP OPERATOR ACTION

001506  000004  017244          GETADR: TYPE,   MSGADR          ;TYPE "FIRST DJ11 ADDRESS"
001512  004537  016215                  JSR     R5,     READIN  ;READ INPUT FROM TTY AND SAVE
001516  001230                          .WORD   DEVADR          ; IN DEVADR
001520  001372                          BNE     GETADR          ;BRANCH IF BAD INPUT
001522  005737  001230                  TST     DEVADR
001526  001003                          BNE     1$
001530  013737  001312  001230          MOV     DEFADR,DEVADR
001536  042737  000007  001230  1$:     BIC     #7,DEVADR
001544  022737  160000  001230          CMP     #160000,DEVADR
001552  101355                          BHI     GETADR

001554  000004  017274          GETVEC: TYPE,   MSGVEC          ;TYPE "VECTOR ADDRESS:"
001560  004537  016216                  JSR     R5,     READIN  ;READ INPUT FROM TTY AND SAVE
001564  001232                          .WORD   VECADR          ;IN VECADR
001566  001372                          BNE     GETVEC          ;BRANCH IF BAD INPUT
001570  005737  001232                  TST     VECADR          ;CHECK FOR CR
001574  001003                          BNE     1$
001576  012737  000300  001232          MOV     #300,   VECADR  ;SET TO FIRST FLOATING VECTOR
001604  042737  000007  001232  1$:     BIC     #7,     VECADR  ;CLEAR TO MODULO 10
001612  022737  000300  001232          CMP     #300,   VECADR  ;CHECK FOR LOW LIMIT
001620  003355                          BGT     GETVEC
001622  022737  001000  001232          CMP     #1000,  VECADR  ;CHECK FOR UPPER LIMIT
001630  003751                          BLE     GETVEC
001632  000004  017320          GETNUM: TYPE,   MSGNUM  ;TYPE"NUMBER OR UNITS:"
001636  004737  016364                  JSR     PC,READ$        ;READ INPUT FROM THE TTY
                                ; CHECK STRING FOR VALID DIGITS,TERMINATOR,AND 'P'
001642  012702  016512                  MOV     #INPUT,R2       ;POINTS TO INPUT STRING
```

```
                                                    MOVB    (R2)+,R1        ;LOAD 1ST CHAR
                                                    BEQ     1$              ;BRANCH IF IMMEDIATE TERMNATOR
                                                    CMPB    #'P,R1          ;CHECK FOR A P
                                                    BEQ     CONFIG          ;CONFIGURE MODE IF SO
                                                    CMPB    R1,#71          ;MUST BE A VALID ASCII NUMBER
                                                    BHI     GETNUM          ;ELSE RETRY
                                                    SUB     #60,R1          ;STRIP ASCII CODE
                                                    BLE     GETNUM          ;SHOULDN'T BE ZERO OR NEG
                                                ;1ST CHAR IS A VALID DIGIT, 1 THRU 9 - CHECK REST OF STRING
                                                    TSTB    (R2)+           ;2ND CHAR A TERMINATOR?
                                                    BEQ     3$              ;YES - R1 HAS THE FINAL # OF UNITS
                                                    TSTB    (R2)            ;NO - NXT CHAR MUST BE THE TERMINATOR
                                                    BNE     GETNUM          ;ELSE RETRY.
                                                    CMPB    -(R2),#71       ;CHECK 2ND CHAR FOR A VALID NUMBER
                                                    BHI     GETNUM
                                                    MOVB    (R2),R3         ;MAKE IT MORE ACCESSIBLE
                                                    SUB     #60,R3          ;STRIP ASCII CODE
                                                    BMI     GETNUM          ;SHOULDN'T BE NEGATIVE
                                                ;BOTH LOW AND HIGH ORDER DIGITS ARE OK - CONVERT DECIMAL VALUE.
                                                    MOV     R1,-(SP)        ;SAVE IT FOR LATER
                                                    ASL     R1              ;MULT HI ORDER BY 8
                                                    ASL     R1
                                                    ASL     R1
                                                    ASL     (SP)            ;MULTIPLY  IT BY 2
                                                    ADD     (SP)+,R1        ;RESULT IS (HI ORD)*10.
                                                    ADD     R3,R1           ;NOW ADD IN THE LOW ORDER
                                                    BR      3$
                                                1$: MOV     #1,R1           ;LOAD DEFAULT VALUE HERE
                                                ;R1 HAS THE CONVERTED VALUE - COMPARE AGAINST MAX ALLOWED.
                                                3$: CMP     R1,#DJMXNO      ;TOO BIG?
                                                    BHI     GETNUM          ;YES - RETRY.
                                                    MOV     R1,UNITS        ;VALUE OK - STORE RESULT
                                                    MOV     #100,DJUUT      ;"PRIME" UNIT UNDER TEST NUMBER
                                                ;
                                                CONFIG: TYPE.   MSGCON          ;TYPE "STANDARD CONFIG?"
                                                    JSR     PC,     READ$   ;READ INPUT FROM TTY
                                                    CMPB    #'P,    INPUT   ;CHECK FOR "P"
                                                    BEQ     RESTAR          ;BRANCH IF PREVIOUS
                                                    MOV     #22.,   R0      ;SET UP COUNTER
                                                    MOV     #LENGTH,R1      ;POINT TO CHAR LEN TABLE
                                                1$: CLR     (R1)+           ;PUT CHAR MASK FOR 9 IN CHAR TABLE
                                                                            ;AND CLR PARITY TABLE
                                                    DEC     R0              ;COUNT DOWN
                                                    BNE     1$              ;BRANCH IF NOT DONE
                                                    TSTB    INPUT           ;CHECK FOR CR
                                                    BEQ     RESTAR          ;BRANCH IF DEFAULT
                                                    CMPB    #131,   INPUT   ;CHECK FOR "Y"
                                                    BEQ     RESTAR          ;BRANCH IF DEFAULT
                                                    CMPB    #116,   INPUT   ;CHECK FOR "N"
                                                    BNE     CONFIG          ;BRANCH IF ILLEGAL ENTRY
                                                    CLR     R0              ;CLR UNIT COUNTER
                                                    CLR     R1              ;CLR LINE COUNTER
                                                    MOV     #LENGTH,R2      ;SET UP POINTER TO CHAR MASK TABLE
                                                    MOV     #PARITY,R3      ;SET UP POINTER TO PARITY TABLE
                                                    MOV     #1,R4           ;SET UP MARKER
                                                TYPLIN: TYPE.   MSGLIN          ;TYPE "LINES "
```

```
                                                    ;TYPE R1 IN OCTAL
                                                    ;AND SUPRESS LEADING ZERO'S
                                                    ;TYPE A DASH (-)
                                                    ;LAST LINE IN GROUP OF 4
                                                    ;TYPE R1 IN OCTAL
                                                    ;AND SUPRESS LEADING ZERO'S
                                                    ;FIRST LINE NEXT GROUP
                                                    ;TYPE "CHAR LENGTH:  "
                                                    ;READ FROM THE TTY
                                                    ;CHECK FOR CR (DEFAULT)
                                                    ;BRANCH IF DEFAULT
                                                    ;CHECK FOR BAD DATA
                                                    ;BRANCH IF BAD
                                                    ;CONVERT FROM ASCII
                                                    ;SET UP MASK
                                                    ;CLEAR MASK BIT BY BIT
                                                    ;BAD INPUT, > 8
                                                    ;COUNT
                                                    ;BRANCH IF NOT DONE
                                                    ;CHECK FOR 5 CHAR
                                                    ;BAD INPUT, < 5
                                                    ;INC TO NEXT LINE GROUP
                                                    ;TYPE "PARITY (NO, ODD, EVEN):
                                                    ;READ INPUT FROM TYY
                                                    ;CHECK FOR CR
                                                    ;BRANCH IF DEFAULT
                                                    ;CHECK FOR "N"
                                                    ;BRANCH IF "NO"
                                                    ;CHECK FOR "E"
                                                    ;BRANCH IF "EVEN"
                                                    ;CHECK FOR "O"
                                                    ;BRANCH IF NONE
                                                    ;SET THE ODD PARITY FLAG
                                                    ;UP DATE PARITY TABLE POINTER
                                                    ;CHECK FOR NEXT UNIT
                                                    ;BRANCH IF NOT
                                                    ;RESET PARITY TABLE POINTER
                                                    ;FLAG TO NEXT UNIT
                                                    ;COUNT UNITS
                                                    ;CHECK FOR LAST
                                                    ;BRANCH IF MORE
```

```
                                                                                
                                                                                
002302  000005                          RESTAR: RESET                           ;ISSUE RESET
002304  012706  001200                          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,E64
                                ;*************************************************************************

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  052505  175510          MOV     #052505,@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,E21,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
```

```
003642  104001                  HLT+1                   ;BIT(S) OF TCR FAILED TO CLEAR
003644  104400                  SCOPE

                        ;*****************************************************************
                        ; TEST 15:       TEST THAT TCR RESPONDS PROPERLY TO BYTE COMMANDS
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-3) E41
                        ;*****************************************************************

003646  012737  177777  175340  TST15:  MOV     #177777,@TCR    ;SET TEST NUMBER IN TCR
003652  105037  175340          CLRB    @TCR            ;CLR EVEN BYTE
003656  016701  175334          MOV     @TCR,   R1      ;SAVE TCR
003662  026127  177400  177400  CMP     #177400,R1      ;CHECK TCR
003670  001401                  BEQ     .+4             ;BRANCH IF OK
003672  104001                  HLT+1                   ;EVEN BYTE CLR FAILED ON TCR
                                                        ;R1 = CONTENTS OF TCR
003674  012737  177777  175318          MOV     #177777,@TCR    ;SET TEST NUMBER IN TCR
003702  005237  175312          INC     TCR             ;INC TO ODD BYTE
003706  105037  175310          CLRB    @TCR            ;CLR ODD BYTE
003712  005237  175306          INC     TCR             ;RESTORE TO EVEN
003716  016701  175300          MOV     @TCR,   R1      ;SAVE TCR
003722  026127  000377  000377  CMP     #000377,R1      ;CHECK TCR
003730  001401                  BEQ     .+4             ;BRANCH IF OK
003732  104001                  HLT+1                   ;ODD BYTE CLR FAILED ON TCR
                                                        ;R1 = CONTENTS OF TCR

003734  104400                  SCOPE

                        ;*****************************************************************
                        ;TEST 16:       TEST THAT THE BIS AND BIC INSTRUCTIONS SET AND CLEAR R/W
                        ;               BITS OF TCR
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-3) E41
                        ;*****************************************************************

003736  005037  175264  TST16:  CLR     @TCR            ;CLEAR THE TCR
003742  016701  175260          MOV     @TCR,   R1      ;CHECK AND SAVE TCR
003746  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
003750  104001                  HLT+1                   ;RESET FAILED TO CLR TCR

003752  052737  177777  175236  BIS     #177777,        @TCR    ;SET ALL R/W BITS OF TCR
003760  026127  177777  177777  CMP     #177777,        @TCR    ;CHECK THAT THEY GOT SET
003766  001401                  BEQ     .+4             ;BRANCH IF OK
003770  104000                  HLT                     ;REG FAILED CMP

003772  042737  177777  175216  BIC     #177777,        @TCR    ;CLEAR TCR
004000  016701  175212          MOV     @TCR,   R1      ;CHECK AND SAVE TCR
004004  001401                  BEQ     .+4             ;BRANCH IF CLEARED OK
004006  104001                  HLT+1                   ;CLR FAILED TO CLR TCR

004010  104400                  SCOPE
004012  012737  000001  001308  MOV     #1,     TIMER   ;INITIALIZE TIMER
004020  012737  004024  001AB8  MOV     #.+6,   LAD     ;RESET LOOP ADDRESS

                        ;*****************************************************************
                        ;TEST 17:       TEST THAT TRANSMIT READY (BITS OF CSR) SETS AND CLEARS
                        ;               WHEN TCRO IS SET AND CLEARED.
```

```
                        ;                       ALSO CHECK THAT THE RIGHT LINE NO. (0) APPEARS IN TBUFF.
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-5) ALL, (D2-6) E23, E32, E33, E49, (D2-8) E3, E1
                        ;*************************************************************************

004024  012777  000400  175156  TST17:  MOV     #400,   @CSR    ;SET XMTR SCAN ENABLE
004032  052777  000001  175150           BIS     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE 0
004040  005002                           CLR     R2              ;SET UP WAIT COUNTER
004042  017701  175142          1$:     MOV     @CSR,   R1      ;CHECK AND SAVE XMTR READY
004046  100404                           BMI     2$              ;BRANCH IF SET OK
004050  005302                           DEC     R2              ;WAIT A WHILE
004052  001373                           BNE     1$
004054  104001                           HLT+1                   ;XMTR READY FAILED TO SET

004056  000416                           BR      3$              ;SKIP LINE # CHECK ON ERROR

004060  052737  001300          2$:     BIS     #0,     @TIMER  ;SET UP TIMER
004064  017701  175126                   MOV     @TBUF,  R1      ;SAVE LINE NUMBER
004070  100001                           BPL     .+4             ;WAIT FOR DONE
004072  000301                           SWAB    R1
004074  042701  000000                   BIC     #BIT0,  R1      ;CHECK THAT THE XMTR STOPPED ON LINE 0
004100  001401                           BEQ     .+4             ;BRANCH IF OK
004102  104001                           HLT+1                   ;WRONG LINE NUMBER APPEARED IN TBUF

004104  017702  175100                   MOV     @CSR,   R2      ;CHECK AND SAVE XMTR READY
004110  100401                           BMI     .+4             ;BRANCH IF OK
004112  104002                           HLT+2                   ;READING THE TBUF CLEARED XMTR READY

004114  042777  000001  175072  3$:     BIC     #BIT0,  @TCR    ;CLR XMTR CONTROL BIT, LINE 0
004122  017702  175068                   MOV     @CSR,   R2      ;CHECK AND SAVE XMTR READY
004126  100401                           BMI     .+4             ;BRANCH IF OK
004130  104001                           HLT+1                   ;XMTR READY FAILED TO CLEAR WHEN
                                        ;       XMTR CONTROL FOR LINE 0 WAS CLEARED

004132  104400                           SCOPE


                        ;*************************************************************************
                        ;TEST 20:       TEST THAT TRANSMIT READY (BIT15 OF CSR) SETS AND CLEARS
                        ;               WHEN EACH TCR BIT IS SET AND CLEARED.
                        ;               ALSO CHECK THAT THE RIGHT LINE NO. APPEARS IN TBUFF.
                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-5) ALL, (D2-6) E23, E32, E33, E49, (D2-8) E3, E1
                        ;*************************************************************************

004134  012777  000400  175046  TST20:  MOV     #400,   @CSR    ;SET XMTR SCAN ENABLE
004142  005001                           CLR     R1
004144  012704  000001                   MOV     #1,     R4      ;SET UP MARKER
004150  050477  175040          4$:     BIS     R4,     @TCR    ;SET XMTR CONTROL BIT, EACH LINE
004154  005002                           CLR     R2              ;SET UP WAIT COUNTER
004156  017702  175026          1$:     MOV     @CSR,   R2      ;CHECK AND SAVE XMTR READY
004162  100404                           BMI     2$              ;BRANCH IF SET OK
004164  005302                           DEC     R2              ;WAIT A WHILE
004166  001373                           BNE     1$
004170  104002                           HLT+2                   ;XMTR READY FAILED TO SET
                                                                ;R1=LINE #
                                                                ;R2=CSR
```

```
1160  004172  000414                        BR      3$              ;SKIP LINE # CHECK ON ERROR
1161
1162  004174  050037  001302          2$:   BIS     R0,     @#TIMER ;SET UP TIMER
1163  004200  016705  176512                MOV     @#TBUF, R2      ;SAVE LINE NUMBER
1164  004204  000302                        SWAB    R2
1165  004206  020201                        CMP     R2,     R1      ;CHECK THAT THE XMTR STOPPED ON RIGHT LINE
1166  004210  001401                        BEQ     .+4             ;BRANCH IF OK
1167  004212  104002                        HLT+2                   ;WRONG LINE NUMBER APPEARED IN TBUF
1168                                                                ;R1=LINE #(SHOULD BE)
1169                                                                ;R2=LINE # (TBUF)
1170
1171  004214  017703  174770                MOV     @#CSR,  R3      ;CHECK AND SAVE XMTR READY
1172  004220  100401                        BMI     .+4             ;BRANCH IF OK
1173  004222  104003                        HLT+3                   ;READING THE TBUF CLEARED XMTR READY
1174
1175  004224  040427  174764          3$:   BIC     R4,     @TCR    ;CLR XMTR CONTROL BIT, EACH LINE
1176  004230  017703  174754                MOV     @#CSR,  R2      ;CHECK AND SAVE XMTR READY
1177  004234  100001                        BPL     .+4             ;BRANCH IF OK
1178  004236  104002                        HLT+2                   ;XMTR READY FAILED TO CLEAR WHEN
1179                                                                ;   XMTR CONTROL FOR LINE .LINE WAS CLEARED
1180                                                                ;R1 = LINE #
1181                                                                ;R2 = CONTENTS OF CSR
1182
1183  004240  005201                        INC     R1              ;INC LINE COUNTER TO NEXT LINE
1184  004242  006304                        ASL     R4              ;SHIFT MARKER TO NEXT LINE
1185  004244  103341                        BCC     4$              ;BRANCH IF NOT LAST LINE
1186
1187  004246  104400                        SCOPE
1188
1189
1190                                        ;*************************************************************************
1191                                        ;TEST 21:       TEST THAT MASTER TRANSMIT SCAN ENABLE (CSR BIT 8) ON A D
1192                                        ;               DISABLES TRANSMITTER READY (CSR BIT 15)
1193                                        ;               WHEN TCR BIT, LINE 0 IS SET.
1194                                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-6) E49,E23,E32,E33
1195                                        ;*************************************************************************
1196
1197  004250  005027  174734        TST21:  CLR     @#CSR           ;CLEAR CONTROL STATUS
1198  004254  052777  000001  174732        BIS     #BIT0,@#TCR     ;SET XMTR CONTROL BIT, LINE 0
1199  004260  013700  001302                MOV     TIMER,R0        ;GET TIMER FROM PREVIOUS TEST
1200  004264  017701  174716          1$:   MOV     @#CSR,R1        ;CHECK AND SAVE XMTR READY
1201  004270  100002                        BPL     2$              ;BRANCH IF NOT SET
1202  004272  104001                        HLT+1                   ;XMTR READY SET WITHOUT
1203  004274  000402                        BR      3$              ;MASTER TRANS SCAN ENABLE
1204  004276  005300                  2$:   DEC     R0              ;WAIT A WHILE
1205  004300  001371                        BNE     1$              ;AND CHECK AGAIN
1206
1207  004302  052777  000400  174676  3$:   BIS     #400,@#CSR      ;SET MASTER TRAN SCAN ENABLE
1208  004310  005000                        CLR     R0
1209  004312  017701  174670          4$:   MOV     @#CSR,   R1     ;CHECK AND SAVE XMTR READY
1210  004316  100403                        BMI     5$
1211  004320  005200                        INC     R0
1212  004322  001373                        BNE     4$
1213  004324  104001                        HLT+1                   ;TRAN READY NEVER CAME UP
1214                                                                ;TCR BIT WAS SET FIRST
```

```
                                                                    ;THEN MASTER TRAN SCAN ENABLE
004330  042777  000001  174656  5$:     BIC     #BIT0,  @TCR       ;CLEAR IT OUT
004336  017701  174646              MOV     @CSR,   R1              ;CHECK AND SAVE TRAN READY
004342  100001                      BPL     .+4                     ;BRANCH IF OK
004344  104001                      HLT+1                           ;TRAN READY DIDN'T CLEAR.

004346  104400                      SCOPE


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

004350  012777  004424  174646  TST22:  MOV     #ISR22,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
004356  012777  000340  174642          MOV     #340,@XMTLVL    ;AT LEVEL 7
004364  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
004372  052737  000340  177776          BIS     #340,@#PS       ;SET PS TO LEVEL 7
004400  012777  040400  174608          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
004406  012777  000001  174600          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
004414  017701  174570          1$:     MOV     @CSR,   R1      ;WAIT
004420  100375                      BPL     1$
004422  000404                      BR      END22           ;OK, BRANCH IF NO INTERRUPT

004424  104000          ISR22:  HLT                             ;SHOULDN'T HAVE INTERRUPTED AT LEVEL 7
004426  012716  004434              MOV     #END22,(SP)     ;MOVE NEW RTI ADR ONTO STACK
004432  000002                      RTI

004434  013777  001226  174562  END22:  MOV     XMTLVL,@XMTVEC
004442  012777  000004  174556          MOV     #IOT,   @XMTLVL
004450  005077  174540                  CLR     @TCR
004454  005077  174530                  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  004554              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                                  ;TEST 24:       TEST THAT INTERRUPT DOES NOT OCCUR AT LEVEL 5
1285                                  ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
1286                                  ;*************************************************************************
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                                  ;TEST 25:       TEST THAT INTERRUPT OCCURS AT LEVEL 4
1313                                  ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
1314                                  ;*************************************************************************
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
1329
```

```
 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
 1335
 1336  005046  104400                          SCOPE
 1337
 1338
 1339                                  ;*************************************************************************
 1340                                  ;TEST 26:       TEST THAT INTERRUPT OCCURS AT LEVEL 3
 1341                                  ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
 1342                                  ;*************************************************************************
 1343
 1344  005050  012777  005126  174146  TST26:  MOV     #ISR26,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
 1345  005056  012777  000340  174142          MOV     #340,@XMTLVL    ;AT LEVEL 7
 1346  005064  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
 1347  005072  052737  000140  177776          BIS     #140,@#PS       ;SET PS TO LEVEL 3
 1348  005100  012777  040400  174102          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
 1349  005106  012777  000001  174100          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
 1350  005114  017701  174070          1$:     MOV     @CSR,   R1      ;WAIT
 1351  005120  100375                          BPL     1$
 1352  005122  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 3
 1353  005124  000403                          BR      END26           ;CONTINUE
 1354
 1355  005126  012716  005134          ISR26:  MOV     #END26,(SP)     ;MOVE NEW RTI ADR ONTO STACK
 1356  005132  000002                          RTI
 1357
 1358  005134  013777  001226  174062  END26:  MOV     XMTLVL,@XMTVEC
 1359  005142  012777  000004  174056          MOV     #IOT,   @XMTLVL
 1360  005150  005077  174040                  CLR     @TCR
 1361  005154  005077  174030                  CLR     @CSR
 1362  005160  042737  000340  177776          BIC     #340,@#PS
 1363
 1364  005166  104400                          SCOPE
 1365
 1366
 1367
 1368                                  ;*************************************************************************
 1369                                  ;TEST 27:       TEST THAT INTERRUPT OCCURS AT LEVEL 2
 1370                                  ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
 1371                                  ;*************************************************************************
 1372
 1373  005170  012777  005246  174026  TST27:  MOV     #ISR27,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
 1374  005176  012777  000340  174022          MOV     #340,@XMTLVL    ;AT LEVEL 7
 1375  005204  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
 1376  005212  052737  000100  177776          BIS     #100,@#PS       ;SET PS TO LEVEL 2
 1377  005220  012777  040400  173762          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
 1378  005226  012777  000001  173760          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
 1379  005234  017701  173750          1$:     MOV     @CSR,   R1      ;WAIT
 1380  005240  100375                          BPL     1$
 1381  005242  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 2
 1382  005244  000403                          BR      END27           ;CONTINUE
 1383
 1384  005246  012716  005254          ISR27:  MOV     #END27,(SP)     ;MOVE NEW RTI ADR ONTO STACK
 1385  005252  000002                          RTI
```

```
1386  005254  013777  001226  173742  END27:  MOV     XMTLVL,@XMTVEC
1387  005262  012777  000004  173736          MOV     #IOT,   @XMTLVL
1388  005270  005077  173720                  CLR     @TCR
1389  005274  005077  173710                  CLR     @CSR
1390  005300  042737  000340  177776          BIC     #340,@#PS
1391
1392  005306  104400                          SCOPE
1393
1394
1395
1396                          ;*************************************************************************
1397                          ;TEST 30:       TEST THAT INTERRUPT OCCURS AT LEVEL 1
1398                          ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
1399                          ;*************************************************************************
1400
1400  005310  012777  005366  173706  TST30:  MOV     #ISR30,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
1401  005316  012777  000340  173702          MOV     #340,@XMTLVL    ;AT LEVEL 7
1402  005324  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
1403  005332  052737  000040  177776          BIS     #040,@#PS       ;SET PS TO LEVEL 1
1404  005340  012777  040400  173642          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
1405  005346  012777  000001  173640          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
1406  005354  017701  173630          1$:     MOV     @CSR,   R1      ;WAIT
1407  005360  100375                          BPL     1$
1408  005362  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 1
1409  005364  000403                          BR      END30           ;CONTINUE
1410
1411  005366  012716  005374          ISR30:  MOV     #END30,(SP)     ;MOVE NEW RTI ADR ONTO STACK
1412  005372  000002                          RTI
1413
1414  005374  013777  001226  173622  END30:  MOV     XMTLVL,@XMTVEC
1415  005402  012777  000004  173616          MOV     #IOT,   @XMTLVL
1416  005410  005077  173600                  CLR     @TCR
1417  005414  005077  173570                  CLR     @CSR
1418  005420  042737  000340  177776          BIC     #340,@#PS
1419
1420  005426  104400                          SCOPE
1421
1422
1423
1424                          ;*************************************************************************
1425                          ;TEST 31:       TEST THAT INTERRUPT OCCURS AT LEVEL 0
1426                          ;PROBABLE FAULTY LOGIC:  M7821 WIRING, PROPER PRIORITY CHIP
1427                          ;*************************************************************************
1428
1428  005430  012777  005506  173566  TST31:  MOV     #ISR31,@XMTVEC  ;SET UP XMTR INTERRUPT VECTOR
1429  005436  012777  000340  173562          MOV     #340,@XMTLVL    ;AT LEVEL 7
1430  005444  042737  000340  177776          BIC     #340,@#PS       ;CLEAR PS LEVEL
1431  005452  052737  000000  177776          BIS     #000,@#PS       ;SET PS TO LEVEL 0
1432  005460  012777  040400  173522          MOV     #040400,@CSR    ;SET TRAN MASTER INT. ENABLE
1433  005466  012777  000001  173520          MOV     #BIT0,  @TCR    ;SET TRAN CONTROL BIT, LINE 0
1434  005474  017701  173510          1$:     MOV     @CSR,   R1      ;WAIT
1435  005500  100375                          BPL     1$
1436  005502  104000                          HLT                     ;SHOULD HAVE INTERRUPTED AT LEVEL 0
1437  005504  000403                          BR      END31           ;CONTINUE
1438
1439  005506  012716  005514          ISR31:  MOV     #END31,(SP)     ;MOVE NEW RTI ADR ONTO STACK
1440  005512  000002                          RTI
1441
```

```
1442  005514  013777  001226  173502  END31:  MOV     XMTLVL,@XMTVEC
1443  005522  012777  000004  173476          MOV     #IQT,   @XMTLVL
1444  005530  005077  173460                  CLR     @TCR
1445  005534  005077  173450                  CLR     @CSR
1446  005540  042737  000340  177776          BIC     #340,@#PS
1447
1448  005546  104400                          SCOPE
1449
1450  005550  005037  001302                  CLR     TIMER           ;INITIALIZE TIMER
1451  005554  012737  005562  015770          MOV     #.+6,   LAD     ;RESET LOOP ADDRESS
1452
1453                        ;*****************************************************************
1454                        ;TEST 32:       TEST THAT LINE 0 CAN TRANSMIT AND
1455                        ;               RECEIVE A CHARACTER. (377)
1456                        ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
1457                        ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
1458                        ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
1459                        ;       4$:  CHECKS THAT RIGHT LINE # (0) IN FI/FO
1460                        ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
1461                        ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
1462                        ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
1463                        ;       8$:  CHECKS THAT DONE CLEARS
1464                        ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
1465                        ;*****************************************************************
1466
1467                        ;               INITIALIZE
1468                        ;       DEVICE"CSR"REGISTER
1469                        ;                                               SET:
1470  005562  004737  015566          TST32:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
1471                                                                  ;BIT3 = CLEAR MOS
1472                                                                  ;BIT8 = MASTER XMTR SCAN ENB
1473                        ;WAIT FOR MOS TO CLEAR
1474                        ;                                       SET:
1475                        ;                                       ;BIT0 = RECEIVER ENABLE
1476                        ;
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  173366          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$:     BIS     R0,     TIMER   ;SAVE TIMER
1492  005642  017701  173344                  MOV     @RBUF,  R1      ;READ THE FI/FO
1493  005646  100401                  BMI     .+4                     ;BRANCH IF CHARACTER READY
1494  005650  104001                  HLT+1                           ;CHARACTER READY DIDN'T SET
1495                                                                  ;R1 = CONTENTS OF RBUF
1496  005652  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
1497  005656  001401                  BEQ     .+4                     ;BRANCH IF NONE
```

```
1498  005660  104001                          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 FI/FO
1516                                                                  ;R2=MASK (BITS SET NOT EXPECTED)
1517  005714  105102                  6$:     COMB    R2              ;REVERSE THE MASK
1518  005716  120102                          CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
1519  005720  001401                          BEQ     .+4             ;BRANCH IF OK
1520  005722  104002                          HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
1521                                                                  ;R1=DATA FROM FI/FO (COMPLETE WORD)
1522                                                                  ;R2=DATA (LOW BYTE) EXPECTED
1523  005724  017701  173262          7$:     MOV     @RBUF,  R1      ;READ FI/FO
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$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
1541                          ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
1542                          ;       4$:  CHECKS THAT RIGHT LINE # (1) IN FI/FO
1543                          ;       5$:  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 DD3 SERIES
1548                          ;*****************************************************************
1549
1550                          ;               INITIALIZE
1551                          ;       DEVICE"CSR"REGISTER
1552                          ;                                       SET:
1553  005762  004737  015566  TST33:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
```

```
1554                                                                    ;BIT3 = CLEAR MOS
1555                                                                    ;BIT8 = MASTER XMTR SCAN ENB
1556                                    ;WAIT FOR MOS TO CLEAR
1557                                    ;                               SET:
1558                                    ;                               ;BIT0 = RECEIVER ENABLE
1559                                    ;
1560  005766  052777  000002  173220            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  006002  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$:    BIS     R0,    TIMER    ;SAVE TIMER
1575  006042  017701  173144                  MOV     @RBUF, R1       ;READ THE FI/FO
1576  006046  100401                           BMI     .+4             ;BRANCH IF CHARACTER READY
1577  006050  104001                           HLT+1                   ;CHARACTER READY DIDN'T SET
1578                                                                    ;R1 = CONTENTS OF RBUF
1579  006052  032701  070000            3$:    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 FI/FO
1599                                                                    ;R2=MASK (BITS SET NOT EXPECTED)
1600  006114  105102                    6$:    COMB    R2              ;REVERSE THE MASK
1601  006116  120102                           CMPB    R1,    R2       ;CHECK THE ACTURAL DATA
1602  006120  001401                           BEQ     .+4             ;BRANCH IF OK
1603  006122  104002                           HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
1604                                                                    ;R1=DATA FROM FI/FO (COMPLETE WORD)
1605                                                                    ;R2=DATA (LOW BYTE) EXPECTED
1606  006124  017701  173062            7$:    MOV     @RBUF, R1       ;READ FI/FO
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          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
 1611  006140  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
 1612  006144  001401                          BEQ     .+4             ;BRANCH IF OK
 1613  006146  104001                          HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
 1614                                                                  ;R1 = CONTENTS OF CSR
 1615  006150  005077  173040                  CLR     @TCR            ;CLEAR TCR
 1616  006154  005077  173030                  CLR     @CSR            ;CLEAR CSR
 1617  006160  104400                          SCOPE
 1618
 1619                                  ;*****************************************************************
 1620                                  ;TEST 34:       TEST THAT LINE 2 CAN TRANSMIT AND
 1621                                  ;               RECEIVE A CHARACTER. (377)
 1622                                  ;
 1623                                  ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
 1624                                  ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
 1625                                  ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
 1626                                  ;       4$:  CHECKS THAT RIGHT LINE # (2) IN FI/FO
 1627                                  ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
 1628                                  ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
 1629                                  ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
 1630                                  ;       8$:  CHECKS THAT DONE CLEARS
 1631                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DD3 SERIES
 1632                                  ;*****************************************************************
 1633
 1634                                  ;               INITIALIZE
 1635                                  ;       DEVICE"CSR"REGISTER
 1636                                  ;                                               SET:
 1637  006162  004737  015566          TST34:  JSR     PC,     @#INITD :BIT2 = MAINTENANCE
 1638                                                                  ;BIT3 = CLEAR MOS
 1639                                                                  ;BIT8 = MASTER XMTR SCAN ENB
 1640                                  ;WAIT FOR MOS TO CLEAR
 1641                                  ;                               SET:
 1642                                  ;                               ;BIT0 = RECEIVER ENABLE
 1643  006166  052777  000004  173020          BIS     #BIT2,  @TCR    ;SET XMTR CONTROL BIT, LINE 2
 1644  006174  017701  173010          LOP34:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 1645  006200  100375                          BPL     LOP34
 1646  006202  012777  000377  173006          MOV     #377,   @TBUF   ;SEND A RUBOUT
 1647  006210  005000                          CLR     R0              ;CLEAR COUNTER
 1648  006212  105305                  1$:     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
 1656                                                                  ;R1 = CONTENTS OF CSR
 1657  006236  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
 1658  006242  017701  172744                  MOV     @RBUF,  R1      ;READ THE FI/FO
 1659  006246  100401                          BMI     .+4             ;BRANCH IF CHARACTER READY
 1660  006250  104001                          HLT+1                   ;CHARACTER READY DIDN'T SET
 1661                                                                  ;R1 = CONTENTS OF RBUF
 1662  006252  032701  070000          3$:     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$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
006264  042702  170377                BIC     #170377,R2      ;MASK LINE#
006270  000302                        SWAB    R2              ;LINE # IN LOW BYTE
006272  122702  000002                CMPB    #2.,    R2      ;CHECK LINE #
006276  001401                        BEQ     .+4             ;BRANCH IF OK
006300  104001                        HLT+1                   ;WRONG LINE # RECEIVED
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BITS8-11 = LINE #
006302  117702  173000        5$:     MOVB    @DJLEN, 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 FI/FO
                                                                ;R2=MASK (BITS SET NOT EXPECTED)
006314  105102                6$:     COMB    R2              ;REVERSE THE MASK
006316  120102                        CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
006320  001401                        BEQ     .+4             ;BRANCH IF OK
006322  104002                        HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                                ;R2=DATA (LOW BYTE) EXPECTED
006324  017701  172662        7$:     MOV     @RBUF,  R1      ;READ FI/FO
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     @CSR,   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     @CSR            ;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 FI/FO
                        ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
                        ;       4$:  CHECKS THAT RIGHT LINE # (3) 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:
006362  004737  015566  TST35:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
```

```
                                        ;WAIT FOR MOS TO CLEAR
                                        ;                                       SET:
                                        ;                                       ;BIT0 = RECEIVER ENABLE
                                        ;
1726  006366  052777  000010  172620          BIS     #BIT3,  @TCR    ;SET XMTR CONTROL BIT, LINE 3
1727  006374  017701  172610        LOP35:    MOV     @CSR,   R1      ;WAIT FOR XMTR READY
1728  006400  100375                         BPL     LOP35
1729  006402  012777  000377  172606          MOV     #377,   @TBUF   ;SEND A RUBOUT
1730  006410  005000                         CLR     R0              ;CLEAR COUNTER
1731  006412  105305                1$:      DECB    R5              ;SHORT WAIT LOOP
1732  006414  001376                         BNE     1$
1733  006416  105777  172566                TSTB    @CSR            ;WAIT FOR DONE
1734  006422  100405                         BMI     2$              ;BRANCH WHEN DONE
1735  006424  005200                         INC     R0              ;TIME COUNTER
1736  006426  001371                         BNE     1$              ;BRANCH IF NOT TIME-OUT
1737  006430  017701  172554                MOV     @CSR,   R1      ;SAVE CSR
1738  006434  104001                         HLT+1                   ;DONE NEVER CAME UP
1739                                                                  ;R1 = CONTENTS OF CSR
1740  006436  050037  001302        2$:      BIS     R0,     TIMER   ;SAVE TIMER
1741  006442  017701  172544                MOV     @RBUF,  R1      ;READ THE FI/FO
1742  006446  100401                         BMI     .+4             ;BRANCH IF CHARACTER READY
1743  006450  104001                         HLT+1                   ;CHARACTER READY DIDN'T SET
1744                                                                  ;R1 = CONTENTS OF RBUF
1745  006452  032701  070000        3$:      BIT     #70000, R1      ;CHECK FOR ERROR BITS
1746  006456  001401                         BEQ     .+4             ;BRANCH IF NONE
1747  006460  104001                         HLT+1                   ;ERROR IN RECEIVED CHAR
1748                                                                  ;R1 = CONTENTS OF RBUF
1749                                                                  ;BIT14=UART OVERRUN
1750                                                                  ;BIT13=FRAMING ERROR
1751                                                                  ;BIT12=PARITY ERROR
1752  006462  010102                4$:      MOV     R1,     R2      ;DUPLICATE DATA WORD
1753  006464  042702  170377                BIC     #170377,R2      ;MASK LINE#
1754  006470  000302                         SWAB    R2              ;LINE # IN LOW BYTE
1755  006472  122702  000003                CMPB    #3.,    R2      ;CHECK LINE #
1756  006476  001401                         BEQ     .+4             ;BRANCH IF OK
1757  006500  104001                         HLT+1                   ;WRONG LINE # RECEIVED
1758                                                                  ;R1 = CONTENTS OF RBUF
1759                                                                  ;BITS8-11 => LINE #
1760  006502  117702  172600        5$:      MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
1761  006506  130201                         BITB    R2,     R1      ;CHECK CHAR LENGTH.
1762  006510  001401                         BEQ     .+4             ;BRANCH IF OK
1763  006512  104002                         HLT+2                   ;WRONG CHARACTER LENGTH
1764                                                                  ;R1=DATA FROM FI/FO
1765                                                                  ;R2=MASK (BITS SET NOT EXPECTED)
1766  006514  105102                6$:      COMB    R2              ;REVERSE THE MASK
1767  006516  120102                         CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
1768  006520  001401                         BEQ     .+4             ;BRANCH IF OK
1769  006522  104002                         HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
1770                                                                  ;R1=DATA FROM FI/FO (COMPLETE WORD)
1771                                                                  ;R2=DATA (LOW BYTE) EXPECTED
1772  006524  017701  172462        7$:      MOV     @RBUF,  R1      ;READ FI/FO
1773  006530  100001                         BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
1774  006532  104001                         HLT+1                   ;CHARACTER PRESENT STAYED SET
1775                                                                  ;R1 = CONTENTS OF RBUF
1776  006534  017701  172450        8$:      MOV     @CSR,   R1      ;SAVE THE CSR
1777  006540  022701  100405                CMP     #100405,R1      ;CHECK THE CSR
```

```
                        006644  012401          BEQ     .+4             ;BRANCH IF OK
                        006646  104001          HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                                        ;R1 = CONTENTS OF CSR
                        006650  005037  172440  CLR     @TCR            ;CLEAR TCR
                        006654  005037  172430  CLR     @CSR            ;CLEAR CSR
                        006660  104401          SCOPE
                        006662  005267  001306  INC     DILEN
                        006666  012767  006674  016770  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; LAST CARD DD3 SERIES
                                        ;*****************************************************************

                                        ;       INITIALIZE
                                        ;       DEVICE"CSR"REGISTER
                                                                        ;SET:
006674  004737  016666          TST36:  JSR     PC,     @#INITD         ;BIT2 = MAINTENANCE
                                                                        ;BIT3 = CLEAR MOS
                                                                        ;BIT8 = MASTER XMTR SCAN ENB
                                        ;WAIT FOR MOS TO CLEAR
                                        ;
                                        ;
                                        ;
                                                                        ;SET:
                                                                        ;BIT0 = RECEIVER ENABLE

006700  052777  000020  172406          BIS     #BIT4,  @TCR    ;SET XMTR CONTROL BIT, LINE 4
006706  017701  172376          LOP36:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
006712  100375                  BPL     LOP36
006714  012777  000377  172374          MOV     #377,   @TBUF   ;SEND A RUBOUT
006722  005000                  CLR     R0              ;CLEAR COUNTER
006724  105300          18:     DECB    R0              ;SHORT WAIT LOOP
006726  001376                  BNE     18
006730  105777  172354          TSTB    @CSR            ;WAIT FOR DONE
006734  100405                  BMI     28              ;BRANCH WHEN DONE
006736  005200                  INC     R0              ;TIME COUNTER
006740  001371                  BNE     18              ;BRANCH IF NOT TIME-OUT
006742  017701  172342          MOV     @CSR,   R1      ;SAVE CSR
006746  104001                  HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
006750  050037  001302          28:     BIS     R0,     TIMER   ;SAVE TIMER
006754  017701  172332          MOV     @RBUF,  R1      ;READ THE FIFO
006760  100404                  BMI     .+4             ;BRANCH IF CHARACTER READY
006762  104001                  HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
006764  032701  070000          38:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
006770  001401                  BEQ     .+4             ;BRANCH IF NONE
006772  104001                  HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
```

C04


```
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR
006674  010102          4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
006676  042702  170377          BIC     #170377,R2      ;MASK LINE#
006702  000302                  SWAB    R2              ;LINE # IN LOW BYTE
006704  122702  000004          CMPB    #4.,    R2      ;CHECK LINE #
006710  001401                  BEQ     .+4             ;BRANCH IF OK
006712  104001                  HLT+1                   ;WRONG LINE # RECEIVED
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 = LINE #
006714  117702  172366  5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
006720  130201                  BITB    R2,     R1      ;CHECK CHAR LENGTH.
006722  001401                  BEQ     .+4             ;BRANCH IF OK
006724  104002                  HLT+2                   ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FI/FO
                                                        ;R2=MASK (BITS SET NOT EXPECTED)
006726  105102          6$:     COMB    R2              ;REVERSE THE MASK
006730  120102                  CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
006732  001401                  BEQ     .+4             ;BRANCH IF OK
006734  104002                  HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                        ;R2=DATA (LOW BYTE) EXPECTED
006736  017701  172250  7$:     MOV     @RBUF,  R1      ;READ FI/FO
006742  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
006744  104001                  HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                        ;R1 = CONTENTS OF RBUF
006746  017701  172236  8$:     MOV     @CSR,   R1      ;SAVE THE CSR
006752  022701  100405          CMP     #100405,R1      ;CHECK THE CSR
006756  001401                  BEQ     .+4             ;BRANCH IF OK
006760  104001                  HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                        ;R1 = CONTENTS OF CSR
006762  005077  172226          CLR     @TCR            ;CLEAR TCR
006766  005077  172216          CLR     @CSR            ;CLEAR CSR
006772  104400                  SCOPE

                        ;*****************************************************************
                        ;TEST 37:       TEST THAT LINE 5 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 # (5) 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:  M7295 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
                        ;*****************************************************************

                        ;               INITIALIZE
                        ;       DEVICE"CSR"REGISTER
                                                        SET:
006774  004737  015566  TST37:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
```

```
                                        ;WAIT FOR MOB TO CLEAR
                                        ;
                                        ;
                                        ;                                SET:
                                        ;                                ;BIT0 = RECEIVER ENABLE
007000  052777  000040  172206          BIS     #BIT5,  @TCR     ;SET XMTR CONTROL BIT, LINE 5
007006  017701  172176          LOP37:  MOV     @CSR,   R1       ;WAIT FOR XMTR READY
007012  100375                  BPL     LOP37
007014  012777  000377  172174          MOV     #377,   @TBUF    ;SEND A RUBOUT
007022  005000                  CLR     R0               ;CLEAR COUNTER
007024  105300                  1$:     DECB    R0               ;SHORT WAIT LOOP
007026  001376                  BNE     1$
007030  105777  172154                  TSTB    @CSR             ;WAIT FOR DONE
007034  100405                  BMI     2$               ;BRANCH WHEN DONE
007036  005200                  INC     R0               ;TIME COUNTER
007040  001371                  BNE     1$               ;BRANCH IF NOT TIME-OUT
007042  017701  172142                  MOV     @CSR,   R1       ;SAVE CSR
007046  104001                  HLT+1                    ;DONE NEVER CAME UP
                                        ;R1 = CONTENTS OF CSR
007050  050037  001302          2$:     BIS     R0,     TIMER    ;SAVE TIMER
007054  017701  172132                  MOV     @RBUF,  R1       ;READ THE FI/FO
007060  100401                  BMI     .+4              ;BRANCH IF CHARACTER READY
007062  104001                  HLT+1                    ;CHARACTER READY DIDN'T SET
                                        ;R1 = CONTENTS OF RBUF
007064  032701  070000          3$:     BIT     #70000, R1       ;CHECK FOR ERROR BITS
007070  001401                  BEQ     .+4              ;BRANCH IF NONE
007072  104001                  HLT+1                    ;ERROR IN RECEIVED CHAR
                                        ;R1 = CONTENTS OF RBUF
                                        ;BIT14=UART OVERRUN
                                        ;BIT13=FRAMING ERROR
                                        ;BIT12=PARITY ERROR
007074  010102                  4$:     MOV     R1,     R2       ;DUPLICATE DATA WORD
007076  042702  170377                  BIC     #170377,R2       ;MASK LINE#
007102  000302                  SWAB    R2               ;LINE # IN LOW BYTE
007104  122702  000005                  CMPB    #5.,    R2       ;CHECK LINE #
007110  001401                  BEQ     .+4              ;BRANCH IF OK
007112  104001                  HLT+1                    ;WRONG LINE # RECEIVED
                                        ;R1 = CONTENTS OF RBUF
                                        ;BITS8-11 = LINE #
007114  117702  172166          5$:     MOVB    @DDJLEN, R2      ;GET MASK OF CHARACTER
007120  130201                  BITB    R2,     R1       ;CHECK CHAR LENGTH.
007122  001401                  BEQ     .+4              ;BRANCH IF OK
007124  104002                  HLT+2                    ;WRONG CHARACTER LENGTH
                                        ;R1=DATA FROM FI/FO
                                        ;R2=MASK (BITS SET NOT EXPECTED)
007126  105102                  6$:     COMB    R2               ;REVERSE THE MASK
007130  120102                  CMPB    R1,     R2       ;CHECK THE ACTURAL DATA
007132  001401                  BEQ     .+4              ;BRANCH IF OK
007134  104002                  HLT+2                    ;WRONG CHAR LEN OR DATA ERROR
                                        ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                        ;R2=DATA (LOW BYTE) EXPECTED
007136  017701  172050          7$:     MOV     @RBUF,  R1       ;READ FI/FO
007142  100001                  BPL     .+4              ;BRANCH IF CHAR PRESENT NOT SET
007144  104001                  HLT+1                    ;CHARACTER PRESENT STAYED SET
                                        ;R1 = CONTENTS OF RBUF
007146  017701  172036          8$:     MOV     @CSR,   R1       ;SAVE THE CSR
007152  022701  100405                  CMP     #100405,R1       ;CHECK THE CSR
```

```
 007156  001401                 BEQ     .+4             ;BRANCH IF OK
 007160  104001                 HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                        ;R1 = CONTENTS OF CSR
 007162  005077  172036         CLR     @TCR            ;CLEAR TCR
 007166  005077  172016         CLR     @CSR            ;CLEAR CSR
 007172  104400                 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 (D2-7) ALL; M7279 ALL; UART CARD DD3 SERIES
                        ;*****************************************************************

                        ;               INITIALIZE
                        ;       DEVICE"CSR"REGISTER
                                                        SET:
 007174  004737  015566 TST40:  JSR     PC,     I%INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB

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

 007200  052777  000100  172006         BIS     #BIT6,  @TCR    ;SET XMTR CONTROL BIT, LINE 6
 007206  017701  171776 LOP40:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 007212  100375                 BPL     LOP40
 007214  012777  000377  171774         MOV     #377,   @TBUF   ;SEND A RUBOUT
 007222  005000                 CLR     R0              ;CLEAR COUNTER
 007224  105305         1$:     DECB    R5              ;SHORT WAIT LOOP
 007226  001376                 BNE     1$
 007230  105777  171754         TSTB    @CSR            ;WAIT FOR DONE
 007234  100405                 BMI     2$              ;BRANCH WHEN DONE
 007236  005200                 INC     R0              ;TIME COUNTER
 007240  001371                 BNE     1$              ;BRANCH IF NOT TIME-OUT
 007242  017701  171742         MOV     @CSR,   R1      ;SAVE CSR
 007246  104001                 HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
 007250  050037  001302         BIS     R0,     TIMER   ;SAVE TIMER
 007254  017701  171732 2$:     MOV     @RBUF,  R1      ;READ THE FI/FO
 007260  100401                 BMI     .+4             ;BRANCH IF CHARACTER READY
 007262  104001                 HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
 007264  032701  070000         BIT     #70000, R1      ;CHECK FOR ERROR BITS
 007270  001401                 BEQ     .+4             ;BRANCH IF NONE
 007272  104001                 HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
```

```
                                                                ;BIT12=PARITY ERROR
                                                                ;DUPLICATE DATA WORD
007274  010102          4$:     MOV     R1,     R2              ;MASK LINE#
007276  042702  170377          BIC     #170377,R2              ;LINE # IN LOW BYTE
007302  000302                  SWAB    R2                      ;CHECK LINE #
007304  122702  000006          CMPB    #6.,    R2              ;BRANCH IF OK
007310  001401                  BEQ     .+4                     ;WRONG LINE # RECEIVED
007312  104001                  HLT+1                           ;R1 = CONTENTS OF RBUF
                                                                ;BITS8-11 = LINE #
007314  117702  171766  5$:     MOVB    @DJLEN, R2              ;GET MASK OF CHARACTER
007320  130201                  BITB    R2,     R1              ;CHECK CHAR LENGTH.
007322  001401                  BEQ     .+4                     ;BRANCH IF OK
007324  104002                  HLT+2                           ;WRONG CHARACTER LENGTH
                                                                ;R1=DATA FROM FI/FO
                                                                ;R2=MASK (BITS SET NOT EXPECTED)
007326  105102          6$:     COMB    R2                      ;REVERSE THE MASK
007330  120102                  CMPB    R1,     R2              ;CHECK THE ACTURAL DATA.
007332  001401                  BEQ     .+4                     ;BRANCH IF OK
007334  104002                  HLT+2                           ;WRONG CHAR LEN OR DATA ERROR
                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                                ;R2=DATA (LOW BYTE) EXPECTED
007336  017701  171650  7$:     MOV     @RBUF,  R1              ;READ FI/FO
007342  100001                  BPL     .+4                     ;BRANCH IF CHAR PRESENT NOT SET
007344  104001                  HLT+1                           ;CHARACTER PRESENT STAYED SET
                                                                ;R1 = CONTENTS OF RBUF
007346  017701  171636  8$:     MOV     @CSR,   R1              ;SAVE THE CSR
007352  022701  100405          CMP     #100405,R1              ;CHECK THE CSR
007356  001401                  BEQ     .+4                     ;BRANCH IF OK
007360  104001                  HLT+1                           ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                                ;R1 = CONTENTS OF CSR
007362  005077  171626          CLR     @TCR                    ;CLEAR TCR
007366  005077  171616          CLR     @CSR                    ;CLEAR CSR
007372  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:
007374  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
                                                    :
007400  052777  000200  171606          BIS     #BIT7,  @TCR    ;SET XMTR CONTROL BIT, LINE 7
007406  017701  171576          LOP41:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
007412  100375                  BPL     LOP41
007414  012777  000377  171574          MOV     #377,   @TBUF   ;SEND A RUBOUT
007422  005000                  CLR     R0                      ;CLEAR COUNTER
007424  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
007426  001376                  BNE     1$
007430  105777  171554                  TSTB    @CSR            ;WAIT FOR DONE
007434  100405                  BMI     2$                      ;BRANCH WHEN DONE
007436  005200                  INC     R0                      ;TIME COUNTER
007440  001371                  BNE     1$                      ;BRANCH IF NOT TIME-OUT
007442  017701  171542          MOV     @CSR,   R1              ;SAVE CSR
007446  104001                  HLT+1                           ;DONE NEVER CAME UP
                                                                ;R1 = CONTENTS OF CSR
007450  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
007454  017701  171532          MOV     @RBUF,  R1              ;READ THE FI/FO
007460  100401                  BMI     .+4                     ;BRANCH IF CHARACTER READY
007462  104001                  HLT+1                           ;CHARACTER READY DIDN'T SET
                                                                ;R1 = CONTENTS OF RBUF
007464  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
007470  001401                  BEQ     .+4                     ;BRANCH IF NONE
007472  104001                  HLT+1                           ;ERROR IN RECEIVED CHAR
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BIT14=UART OVERRUN
                                                                ;BIT13=FRAMING ERROR
                                                                ;BIT12=PARITY ERROR
007474  010102                  4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
007476  042702  170377          BIC     #170377,R2              ;MASK LINE#
007502  000302                  SWAB    R2                      ;LINE # IN LOW BYTE
007504  122702  000007          CMPB    #7.,    R2              ;CHECK LINE #
007510  001401                  BEQ     .+4                     ;BRANCH IF OK
007512  104001                  HLT+1                           ;WRONG LINE # RECEIVED
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BITS8-11 = LINE #
007514  117702  171566          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
007520  130201                  BITB    R2,     R1              ;CHECK CHAR LENGTH.
007522  001401                  BEQ     .+4                     ;BRANCH IF OK
007524  104002                  HLT+2                           ;WRONG CHARACTER LENGTH
                                                                ;R1=DATA FROM FI/FO
                                                                ;R2=MASK (BITS SET NOT EXPECTED)
007526  105102                  6$:     COMB    R2              ;REVERSE THE MASK
007530  120102                  CMPB    R1,     R2              ;CHECK THE ACTURAL DATA
007532  001401                  BEQ     .+4                     ;BRANCH IF OK
007534  104002                  HLT+2                           ;WRONG CHAR LEN OR DATA ERROR
                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                                ;R2=DATA (LOW BYTE) EXPECTED
007536  017701  171450          7$:     MOV     @RBUF,  R1      ;READ FI/FO
007542  100001                  BPL     .+4                     ;BRANCH IF CHAR PRESENT NOT SET
007544  104001                  HLT+1                           ;CHARACTER PRESENT STAYED SET
                                                                ;R1 = CONTENTS OF RBUF
007546  017701  171436          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
007552  022701  100405          CMP     #100405,R1              ;CHECK THE CSR
007556  001401                  BEQ     .+4                     ;BRANCH IF OK
007560  104001                  HLT+1                           ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
```

```
                                                        ;R1 = CONTENTS OF CSR
007562  005077  171426          CLR     @TCR            ;CLEAR TCR
007566  005077  171416          CLR     @DCSR           ;CLEAR CSR
007572  104400                  SCOPE
007574  005337  001306          INC     DJLEN
007600  012737  007606  015770  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  015566  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  001302  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
007706  010102                4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
007710  042702  170377                BIC     #170377,R2      ;MASK LINE#
007714  000302                        SWAB    R2              ;LINE # IN LOW BYTE
007716  122702  000010                CMPB    #8.,    R2      ;CHECK LINE #
007722  001401                        BEQ     .+4             ;BRANCH IF OK
007724  104001                        HLT+1                   ;WRONG LINE # RECEIVED
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 = LINE #
007726  117702  171354          5$:     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 FI/FO
                                                        ;R2=MASK (BITS SET NOT EXPECTED)
007740  105102                6$:     COMB    R2              ;REVERSE THE MASK
007742  120102                        CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
007744  001401                        BEQ     .+4             ;BRANCH IF OK
007746  104002                        HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                        ;R2=DATA (LOW BYTE) EXPECTED
007750  017701  171235          7$:     MOV     @RBUF,  R1      ;READ FI/FO
007754  100001                        BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
007756  104001                        HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                        ;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 FI/FO
                        ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
                        ;       4$:  CHECKS THAT RIGHT LINE # (9) 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:
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
                                        ;
2228  010012  052777  001000  171174          BIS     #BIT9,  @TCR    ;SET XMTR CONTROL BIT, LINE 9
2229  010020  017701  171164        LOP43:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
2230  010024  100375                  BPL     LOP43
2231  010026  012777  000377  171162          MOV     #377,   @TBUF   ;SEND A RUBOUT
2232  010034  005000                  CLR     R0                      ;CLEAR COUNTER
2233  010036  105305                1$:       DECB    R5              ;SHORT WAIT LOOP
2234  010040  001376                  BNE     1$
2235  010042  105777  171142          TSTB    @CSR                    ;WAIT FOR DONE
2236  010046  100405                  BMI     2$                      ;BRANCH WHEN DONE
2237  010050  005200                  INC     R0                      ;TIME COUNTER
2238  010052  001371                  BNE     1$                      ;BRANCH IF NOT TIME-OUT
2239  010054  017701  171130          MOV     @CSR,   R1              ;SAVE CSR
2240  010060  104001                  HLT+1                           ;DONE NEVER CAME UP
2241                                                                  ;R1 = CONTENTS OF CSR
2242  010062  050037  001302        2$:       BIS     R0,     TIMER   ;SAVE TIMER
2243  010066  017701  171120          MOV     @RBUF,  R1              ;READ THE FI/FO
2244  010072  100401                  BMI     .+4                     ;BRANCH IF CHARACTER READY
2245  010074  104001                  HLT+1                           ;CHARACTER READY DIDN'T SET
2246                                                                  ;R1 = CONTENTS OF RBUF
2247  010076  032701  070000        3$:       BIT     #70000, R1      ;CHECK FOR ERROR BITS
2248  010102  001401                  BEQ     .+4                     ;BRANCH IF NONE
2249  010104  104001                  HLT+1                           ;ERROR IN RECEIVED CHAR
2250                                                                  ;R1 = CONTENTS OF RBUF
2251                                                                  ;BIT14=UART OVERRUN
2252                                                                  ;BIT13=FRAMING ERROR
2253                                                                  ;BIT12=PARITY ERROR
2254  010106  010102                4$:       MOV     R1,     R2      ;DUPLICATE DATA WORD
2255  010110  042702  170377          BIC     #170377,R2              ;MASK LINE#
2256  010114  000302                  SWAB    R2                      ;LINE # IN LOW BYTE
2257  010116  122702  000011          CMPB    #9.,    R2              ;CHECK LINE #
2258  010122  001401                  BEQ     .+4                     ;BRANCH IF OK
2259  010124  104001                  HLT+1                           ;WRONG LINE # RECEIVED
2260                                                                  ;R1 = CONTENTS OF RBUF
2261                                                                  ;BITS8-11 = LINE #
2262  010126  117702  171154        5$:       MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
2263  010132  130201                  BITB    R2,     R1              ;CHECK CHAR LENGTH.
2264  010134  001401                  BEQ     .+4                     ;BRANCH IF OK
2265  010136  104002                  HLT+2                           ;WRONG CHARACTER LENGTH
2266                                                                  ;R1=DATA FROM FI/FO
2267                                                                  ;R2=MASK (BITS SET NOT EXPECTED)
2268  010140  105102                6$:       COMB    R2              ;REVERSE THE MASK
2269  010142  120102                  CMPB    R1,     R2              ;CHECK THE ACTURAL DATA
2270  010144  001401                  BEQ     .+4                     ;BRANCH IF OK
2271  010146  104002                  HLT+2                           ;WRONG CHAR LEN OR DATA ERROR
2272                                                                  ;R1=DATA FROM FI/FO (COMPLETE WORD)
2273                                                                  ;R2=DATA (LOW BYTE) EXPECTED
2274  010150  017701  171036        7$:       MOV     @RBUF,  R1      ;READ FI/FO
2275  010154  100001                  BPL     .+4                     ;BRANCH IF CHAR PRESENT NOT SET
2276  010156  104001                  HLT+1                           ;CHARACTER PRESENT STAYED SET
2277                                                                  ;R1 = CONTENTS OF RBUF
2278  010160  017701  171024        8$:       MOV     @CSR,   R1      ;SAVE THE CSR
2279  010164  022701  100405          CMP     #100405,R1              ;CHECK THE CSR
2280  010170  001401                  BEQ     .+4                     ;BRANCH IF OK
2281  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
```

```
002338  010310  042702  170377          BIC     #170377,R2      ;MASK LINE#
002339  010314  000302                  SWAB    R2              ;LINE # IN LOW BYTE
002340  010316  122702  000012          CMPB    #10.,   R2      ;CHECK LINE #
002341  010322  001401                  BEQ     .+4             ;BRANCH IF OK
002342  010324  104001                  HLT+1                   ;WRONG LINE # RECEIVED
002343                                                          ;R1 = CONTENTS OF RBUF
002344                                                          ;BITS8-11 = LINE #
002345  010326  117702  170754  5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
002346  010332  130201                  BITB    R2,     R1      ;CHECK CHAR LENGTH.
002347  010334  001401                  BEQ     .+4             ;BRANCH IF OK
002348  010336  104002                  HLT+2                   ;WRONG CHARACTER LENGTH
002349                                                          ;R1=DATA FROM FI/FO
002350                                                          ;R2=MASK (BITS SET NOT EXPECTED)
002351  010340  105102          6$:     COMB    R2              ;REVERSE THE MASK
002352  010342  120102                  CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
002353  010344  001401                  BEQ     .+4             ;BRANCH IF OK
002354  010346  104002                  HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
002355                                                          ;R1=DATA FROM FI/FO (COMPLETE WORD)
002356                                                          ;R2=DATA (LOW BYTE) EXPECTED
002357  010350  017701  170636  7$:     MOV     @RBUF,  R1      ;READ FI/FO
002358  010354  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
002359  010356  104001                  HLT+1                   ;CHARACTER PRESENT STAYED SET
002360                                                          ;R1 = CONTENTS OF RBUF
002361  010360  017701  170624  8$:     MOV     @CSR,   R1      ;SAVE THE CSR
002362  010364  022701  100405          CMP     #100405,R1      ;CHECK THE CSR
002363  010370  001401                  BEQ     .+4             ;BRANCH IF OK
002364  010372  104001                  HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
002365                                                          ;R1 = CONTENTS OF CSR
002366  010374  005077  170614          CLR     @TCR            ;CLEAR TCR
002367  010400  005077  170604          CLR     @CSR            ;CLEAR CSR
002368  010404  104400                  SCOPE
002369
002370                          ;*****************************************************************
002371                          ;TEST 45:       TEST THAT LINE 11 CAN TRANSMIT AND
002372                          ;               RECEIVE A CHARACTER. (CRC
002373                          ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
002374                          ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
002375                          ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
002376                          ;       4$:  CHECKS THAT RIGHT LINE # (11) IN FI/FO
002377                          ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
002378                          ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
002379                          ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
002380                          ;       8$:  CHECKS THAT DONE CLEARS
002381                          ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD D03 SERIES
002382                          ;*****************************************************************
002383
002384                          ;               INITIALIZE
002385                          ;       DEVICE"CSR"REGISTER
002386                                                  SET:
002387  010406  004737  015566  TST45:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
002388                                                          ;BIT3 = CLEAR MOS
002389                                                          ;BIT8 = MASTER XMTR SCAN ENB
002390                          ;WAIT FOR MOS TO CLEAR
002391                                                  SET:
002392                          ;                               ;BIT0 = RECEIVER ENABLE
002393                          ;
```

```
2394  010412  052777  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  170562          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  005077  170414                  CLR     @TCR            ;CLEAR TCR
```

```
002450  010600  005077  170404                  CLR     @CSR            ;CLEAR CSR
002451  010604  104400                          SCOPE
002452  010606  005237  001306                  INC     DJLEN
002453  010612  012737  010620  015770          MOV     #.+6,   LAD
002454
002455                          ;*************************************************************
002456                          ;TEST 46:       TEST THAT LINE 12 CAN TRANSMIT AND
002457                          ;               RECEIVE A CHARACTER. (377)
002458                          ;       1$:  CHECKS THAT DONE SETS IN REASONABLE TIME.
002459                          ;       2$:  CHECKS THAT CHAR PRESENT IS IN FI/FO
002460                          ;       3$:  CHECKS THAT NO ERRORS IN FI/FO
002461                          ;       4$:  CHECKS THAT RIGHT LINE # (12) IN FI/FO
002462                          ;       5$:  CHECKS FOR RIGHT CHARACTER LENGTH
002463                          ;       6$:  CHECKS THAT CORRECT DATA WAS RECEIVED
002464                          ;       7$:  CHECKS THAT CHARACTER PRESENT CLEARS
002465                          ;       8$:  CHECKS THAT DONE CLEARS
002466                          ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) ALL; M7279 ALL; UART CARD DC3 SERIES
002467                          ;*************************************************************
002468
002469                          ;               INITIALIZE
002470                          ;       DEVICE"CSR"REGISTER
002471                                                          SET:
002472  010620  004737  015566  TST46:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
002473                                                          ;BIT3 = CLEAR MOS
002474                                                          ;BIT8 = MASTER XMTR SCAN ENB
002475                          ;WAIT FOR MOS TO CLEAR
002476                          ;
002477                                                          SET:
002478                          ;                               ;BIT0 = RECEIVER ENABLE
002479  010624  052777  010000  170362          BIS     #BIT12, @TCR    ;SET XMTR CONTROL BIT, LINE 12
002480  010632  017701  170352          LOP46:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
002481  010636  100375                          BPL     LOP46
002482  010640  012777  000377  170350          MOV     #377,   @TBUF   ;SEND A RUBOUT
002483  010646  005000                          CLR     R0              ;CLEAR COUNTER
002484  010650  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
002485  010652  001376                          BNE     1$
002486  010654  105777  170330                  TSTB    @CSR            ;WAIT FOR DONE
002487  010660  100405                          BMI     2$              ;BRANCH WHEN DONE
002488  010662  005200                          INC     R0              ;TIME COUNTER
002489  010664  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
002490  010666  017701  170316                  MOV     @CSR,   R1      ;SAVE CSR
002491  010672  104001                          HLT+1                   ;DONE NEVER CAME UP
002492                                                                  ;R1 = CONTENTS OF CSR
002493  010674  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
002494  010700  017701  170306                  MOV     @RBUF,  R1      ;READ THE FI/FO
002495  010704  100401                          BMI     .+4             ;BRANCH IF CHARACTER READY
002496  010706  104001                          HLT+1                   ;CHARACTER READY DIDN'T SET
002497                                                                  ;R1 = CONTENTS OF RBUF
002498  010710  032701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
002499  010714  001401                          BEQ     .+4             ;BRANCH IF NONE
002500  010716  104001                          HLT+1                   ;ERROR IN RECEIVED CHAR
002501                                                                  ;R1 = CONTENTS OF RBUF
002502                                                                  ;BIT14=UART OVERRUN
002503                                                                  ;BIT13=FRAMING ERROR
002504                                                                  ;BIT12=PARITY ERROR
002505  010720  010102          4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
```

```
                                B10     #170377,R2      ;MASK LINE#
                                SWAB    R2              ;LINE # IN LOW BYTE
                                CMPB    #12..   R2      ;CHECK LINE #
                                BEQ     .+4             ;BRANCH IF OK
                                HLT+1                   ;WRONG LINE # RECEIVED
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BITS8-11 = LINE #
                        5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
                                BITB    R2,     R1      ;CHECK CHAR LENGTH.
                                BEQ     .+4             ;BRANCH IF OK
                                HLT+2                   ;WRONG CHARACTER LENGTH
                                                        ;R1=DATA FROM FI/FO
                                                        ;R2=MASK (BITS SET NOT EXPECTED)
                        6$:     COMB    R2              ;REVERSE THE MASK
                                CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
                                BEQ     .+4             ;BRANCH IF OK
                                HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                        ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                        ;R2=DATA (LOW BYTE) EXPECTED
                        7$:     MOV     @RBUF,  R1      ;READ FI/FO
                                BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
                                HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                        ;R1 = CONTENTS OF RBUF
                        8$:     MOV     @CSR,   R1      ;SAVE THE CSR
                                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 47:       TEST THAT LINE 13 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 # (13) 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; M7379 ALL; UART CARD D03 SERIES
;*************************************************************************

;               INITIALIZE
;       DEVICE"CSR"REGISTER
;                                               SET:
D11020  004737  015566  TST47:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
;WAIT FOR MOS TO CLEAR
;                                               SET:
;                                               ;BIT0 = RECEIVER ENABLE
;
```

```
011024  052777  020000  170162          LOP47:  BIS     #BIT13,@TCR     ;SET XMTR CONTROL BIT, LINE 13
011030  017701  170152                  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
011034  100375                          BPL     LOP47
011036  012777  000377  170150          MOV     #377,   @TBUF   ;SEND A RUBOUT
011042  005000                          CLR     R0              ;CLEAR COUNTER
011044  105305                  1$:     DECB    R5              ;SHORT WAIT LOOP
011046  001377                          BNE     1$
011050  105777  170130                  TSTB    @CSR            ;WAIT FOR DONE
011054  100405                          BMI     2$              ;BRANCH WHEN DONE
011056  005200                          INC     R0              ;TIME COUNTER
011060  001371                          BNE     1$              ;BRANCH IF NOT TIME-OUT
011062  017701  170116                  MOV     @CSR,   R1      ;SAVE CSR
011066  104001                          HLT+1                   ;DONE NEVER CAME UP
                                                                ;R1 = CONTENTS OF CSR
011070  050037  001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
011074  017701  170106                  MOV     @RBUF,  R1      ;READ THE FI/FO
011100  100401                          BMI     .+4             ;BRANCH IF CHARACTER READY
011102  104001                          HLT+1                   ;CHARACTER READY DIDN'T SET
                                                                ;R1 = CONTENTS OF RBUF
011104  033701  070000          3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
011110  001401                          BEQ     .+4             ;BRANCH IF NONE
011112  104001                          HLT+1                   ;ERROR IN RECEIVED CHAR
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BIT14=UART OVERRUN
                                                                ;BIT13=FRAMING ERROR
                                                                ;BIT12=PARITY ERROR
011114  010102          4$:     MOV     R1,     R2              ;DUPLICATE DATA WORD
011116  042702  170377          BIC     #^C170377,R2            ;MASK LINE#
011122  000302                  SWAB    R2                      ;LINE # IN LOW BYTE
011124  122702  000015          CMPB    #13.,   R2              ;CHECK LINE #
011130  001401                          BEQ     .+4             ;BRANCH IF OK
011132  104001                          HLT+1                   ;WRONG LINE # RECEIVED
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BITS8-11 = LINE #
011134  117702  170142          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
011140  130201                          BITB    R2,     R1      ;CHECK CHAR LENGTH.
011142  001401                          BEQ     .+4             ;BRANCH IF OK
011144  104002                          HLT+2                   ;WRONG CHARACTER LENGTH
                                                                ;R1=DATA FROM FI/FO
                                                                ;R2=MASK (BITS SET NOT EXPECTED)
011146  105102          6$:     COMB    R2                      ;REVERSE THE MASK
011150  130102                  CMPB    R1,     R2              ;CHECK THE ACTURAL DATA
011152  001401                          BEQ     .+4             ;BRANCH IF OK
011154  104002                          HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                                ;R2=DATA (LOW BYTE) EXPECTED
011156  017701  170024          7$:     MOV     @RBUF,  R1      ;READ FI/FO
011162  100001                          BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
011164  104001                          HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                                ;R1 = CONTENTS OF RBUF
011166  017701  170010          8$:     MOV     @CSR,   R1      ;SAVE THE CSR
011172  022701  100405                  CMP     #100405,R1      ;CHECK THE CSR
011176  001401                          BEQ     .+4             ;BRANCH IF OK
011200  104001                          HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                                ;R1 = CONTENTS OF CSR
011202  005077  170002                  CLR     @TCR            ;CLEAR TCR
```

```
011810  005077  167772          CLR     @CSR            ;CLEAR CSR
011814  104400                  SCOPE

                        ;*************************************************************************
                        ;TEST 50:       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 DD3 SERIES
                        ;*************************************************************************

                        ;               INITIALIZE
                        ;       DEVICE"CSR"REGISTER
                        ;                                               SET:
011820  004737  015566  TST50:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = MASTER XMTR SCAN ENB
                        ;WAIT FOR MOS TO CLEAR
                        ;                               SET:
                        ;                               ;BIT0 = RECEIVER ENABLE
                        ;
011824  052777  040000  167762          BIS     #BIT14, @TCR    ;SET XMTR CONTROL BIT, LINE 14
011832  017701  167752  LOP50:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
011836  100375          BPL     LOP50
011840  012777  000377  167750          MOV     #377,   @TBUF   ;SEND A RUBOUT
011846  005000                  CLR     R0              ;CLEAR COUNTER
011850  105305          1$:     DECB    R5              ;SHORT WAIT LOOP
011852  001376                  BNE     1$
011854  105777  167730          TSTB    @CSR            ;WAIT FOR DONE
011860  100405                  BMI     2$              ;BRANCH WHEN DONE
011862  005200                  INC     R0              ;TIME COUNTER
011864  001371                  BNE     1$              ;BRANCH IF NOT TIME-OUT
011866  017701  167716          MOV     @CSR,   R1      ;SAVE CSR
011872  104001                  HLT+1                   ;DONE NEVER CAME UP
                                                        ;R1 = CONTENTS OF CSR
011874  050037  001302  2$:     BIS     R0,     TIMER   ;SAVE TIMER
012000  017701  167706          MOV     @RBUF,  R1      ;READ THE FI/FO
012004  100401                  BMI     .+4             ;BRANCH IF CHARACTER READY
012006  104001                  HLT+1                   ;CHARACTER READY DIDN'T SET
                                                        ;R1 = CONTENTS OF RBUF
012010  032701  070000  3$:     BIT     #70000, R1      ;CHECK FOR ERROR BITS
012014  001401                  BEQ     .+4             ;BRANCH IF NONE
012016  104001                  HLT+1                   ;ERROR IN RECEIVED CHAR
                                                        ;R1 = CONTENTS OF RBUF
                                                        ;BIT14=UART OVERRUN
                                                        ;BIT13=FRAMING ERROR
                                                        ;BIT12=PARITY ERROR
012020  010102          4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
012022  042702  170377          BIC     #170377,R2      ;MASK LINE#
012026  000302                  SWAB    R2              ;LINE # IN LOW BYTE
```

```
                011330  122702  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  117702  167742  5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
                011344  130201                  BITB    R2,     R1      ;CHECK CHAR LENGTH.
                011346  001401                  BEQ     .+4             ;BRANCH IF OK
                011350  104002                  HLT+2                   ;WRONG CHARACTER LENGTH
                                                                        ;R1=DATA FROM FI/FO
                                                                        ;R2=MASK (BITS SET NOT EXPECTED)
                011352  105102          6$:     COMB    R2              ;REVERSE THE MASK
                011354  130102                  BITB    R1,     R2      ;CHECK THE ACTURAL DATA
                011356  001401                  BEQ     .+4             ;BRANCH IF OK
                011360  104002                  HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                                        ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                                        ;R2=DATA (LOW BYTE) EXPECTED
                011362  017701  167624  7$:     MOV     @RBUF,  R1      ;READ FI/FO
                011366  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
                011370  104001                  HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                                        ;R1 = CONTENTS OF RBUF
                011372  017701  167612  8$:     MOV     @CSR,   R1      ;SAVE THE CSR
                011376  022701  100405          CMP     #100405,R1      ;CHECK THE CSR
                011402  001401                  BEQ     .+4             ;BRANCH IF OK
                011404  104001                  HLT+1                   ;DONE DIDN'T CLEAR OR OTHER CSR ERROR
                                                                        ;R1 = CONTENTS OF CSR
                011406  005077  167602          CLR     @TCR            ;CLEAR TCR
                011412  005077  167572          CLR     @CSR            ;CLEAR CSR
                011416  104400                  SCOPE

                                        ;*************************************************************************
                                        ;TEST 51:       TEST THAT LINE 15 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 # (15) 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:
                011420  004737  015566  TST51:  JSR     PC,     @#INITD ;BIT2 = MAINTENANCE
                                                                        ;BIT3 = CLEAR MOS
                                                                        ;BIT8 = MASTER XMTR SCAN ENB
                                        ;WAIT FOR MOS TO CLEAR
                                                                        ;SET:
                                                                        ;BIT0 = RECEIVER ENABLE

                011424  052777  100000  167562          BIS     #BIT15, @TCR    ;SET XMTR CONTROL BIT, LINE 15
                011432  017701  167552  LOPS1:  MOV     @CSR,   R1      ;WAIT FOR XMTR READY
```

```
                                        BPL     LOP51
                000377  167550          MOV     #377.,  @TBUF   ;SEND A RUBOUT
                                        CLR     R0              ;CLEAR COUNTER
                                1$:     DECB    R5              ;SHORT WAIT LOOP
                                        BNE     1$
                167530                  TSTB    @CSR            ;WAIT FOR DONE
                                        BMI     2$              ;BRANCH WHEN DONE
                                        INC     R0              ;TIME COUNTER
                                        BNE     1$              ;BRANCH IF NOT TIME-OUT
                167516                  MOV     @CSR,   R1      ;SAVE CSR
                                        HLT+1                   ;DONE NEVER CAME UP
                                                                ;R1 = CONTENTS OF CSR
                001302          2$:     BIS     R0,     TIMER   ;SAVE TIMER
                167506                  MOV     @RBUF,  R1      ;READ THE FI/FO
                                        BMI     .+4             ;BRANCH IF CHARACTER READY
                                        HLT+1                   ;CHARACTER READY DIDN'T SET
                                                                ;R1 = CONTENTS OF RBUF
                070000          3$:     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
                                4$:     MOV     R1,     R2      ;DUPLICATE DATA WORD
                170377                  BIC     #170377,R2      ;MASK LINE#
                                        SWAB    R2              ;LINE # IN LOW BYTE
                000017                  CMPB    #15.,   R2      ;CHECK LINE #
                                        BEQ     .+4             ;BRANCH IF OK
                                        HLT+1                   ;WRONG LINE # RECEIVED
                                                                ;R1 = CONTENTS OF RBUF
                                                                ;BITS8-11 = LINE #
                167542          5$:     MOVB    @DJLEN, R2      ;GET MASK OF CHARACTER
                                        BITB    R2,     R1      ;CHECK CHAR LENGTH.
                                        BEQ     .+4             ;BRANCH IF OK
                                        HLT+2                   ;WRONG CHARACTER LENGTH
                                                                ;R1=DATA FROM FI/FO
                                                                ;R2=MASK (BITS SET NOT EXPECTED)
                                6$:     COMB    R2              ;REVERSE THE MASK
                                        CMPB    R1,     R2      ;CHECK THE ACTURAL DATA
                                        BEQ     .+4             ;BRANCH IF OK
                                        HLT+2                   ;WRONG CHAR LEN OR DATA ERROR
                                                                ;R1=DATA FROM FI/FO (COMPLETE WORD)
                                                                ;R2=DATA (LOW BYTE) EXPECTED
                167424          7$:     MOV     @RBUF,  R1      ;READ FI/FO
                                        BPL     .+4             ;BRANCH IF CHAR PRESENT NOT SET
                                        HLT+1                   ;CHARACTER PRESENT STAYED SET
                                                                ;R1 = CONTENTS OF RBUF
                167412          8$:     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
                167402                  CLR     @TCR            ;CLEAR TCR
                167372                  CLR     @CSR            ;CLEAR CSR
                                        SCOPE
```

```
011620  162737  000003  001306          SUB     #3,     DJLEN
011626  005237  001302                  INC     TIMER                   ;WORSE CASE TIME, ONE CHARACTER
011632  013700  001302                  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
                                ;
011662  004737  015566          TST52:  JSR     PC,@#INITD
                                ;SET:
                                ;       BIT2 = MAINTENANCE
                                ;       BIT3 = CLEAR MOS
                                ;       BIT8 = TRANS SCAN ENABLE
                                ;       WAIT FOR BIT4 = MOS CLEAR
                                ;SET:
                                ;       BIT0 = RECEIVER ENABLE
011666  012777  000001  167320          MOV     #BIT0,  @TCR            ;TRANS CONTROL, LINE0
011674  012704  000004                  MOV     #4,R4                   ;SET UP COUNTER
011700  017701  167304          1$:     MOV     @CSR,R1                 ;WAIT FOR TRANS READY
011704  100375                          BPL     1$
011706  010477  167304                  MOV     R4,@TBUF                ;TRANSMIT COUNT
011712  005304                          DEC     R4                      ;COUNT DOWN
011714  001371                          BNE     1$

011716  105777  167266          2$:     TSTB    @CSR                    ;MAKE SURE DONE IS SET
011722  100375                          BPL     2$

011724  052777  000010  167256          BIS     #BIT3,@CSR              ;CLEAR MOS
011732  032777  000020  167250  3$:     BIT     #BIT4,@CSR              ;WAIT FOR CLRMOS TO FINISH
011740  001374                          BNE     3$

011742  017701  167244                  MOV     @RBUF,R1                ;CHECK RBUF AND SAVE
011746  100001                          BPL     .+4
011750  104001                          HLT+1
011752  017701  167232                  MOV     @CSR,R1                 ;SAVE CSR
011756  022701  100405                  CMP     #100405,R1              ;CHECK CSR
011762  001401                          BEQ     .+4                     ;BRANCH IF OK
011764  104001                          HLT+1                           ;CSR ERROR, POSSIBILITIES:
                                                                        ;(1) DONE DIDN'T CLEAR

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

011766  013700  001302                  MOV     TIMER,  R0              ;SET UP TIMER
011772  105300                  4$:     DECB    R5                      ;SHORT WAIT LOOP
011774  001376                          BNE     4$
```

```
 012776  017701  167206                 MOV     @CSR,R1         ;SAVE CSR
 013002  105701                         TSTB    R1              ;CHECK FOR DONE
 013004  100001                         BPL     .+4             ;BRANCH IF OK
 013006  104001                         HLT+1                   ;DONE CAME UP!
                                                                 ;MCS MUST NOT HAVE CLEARED
 013010  005300                         DEC     R0              ;TIMER COUNT
 013012  001367                         BNE     4$

 013014  022701  100405                 CMP     #100405,R1      ;CHECK CSR
 013020  001401                         BEQ     .+4             ;BRANCH IF OK
 013022  104001                         HLT+1                   ;CSR ERROR

 013024  017701  167162                 MOV     @RBUF,R1        ;CHECK RBUF
 013030  100001                         BPL     .+4             ;BRANCH IF OK
 013032  104001                         HLT+1                   ;RBUF NOT EMPTY!

 013034  005077  167154                 CLR     @TCR
 013040  005077  167144                 CLR     @CSR

 013044  104400                         SCOPE


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

                                ;               INITIALIZE
                                ;       DEVICE"CSR"REGISTER
                                ;                                               SET:
 013046  004737  015566         TST53:  JSR     PC,     @#INITD         ;BIT2 = MAINTENANCE
                                                                         ;BIT3 = CLEAR MCS
                                                                         ;BIT8 = MASTER XMTR SCAN ENB
                                ;WAIT FOR MCS TO CLEAR
                                ;                                               SET:
                                ;                                               ;BIT0 = RECEIVER ENABLE
                                ;
 013052  052777  000001  167134         BIS     #BIT0,  @TCR    ;TRANS CONTROL, LINE 0
 013060  017701  167124         1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 013064  100375                 BPL     1$
 013066  012777  000001  167122         MOV     #1,     @TBUF   ;TRANSMIT A 1
 013074  017701  167110         2$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 013100  100375                 BPL     2$
 013102  012777  000002  167106         MOV     #2,     @TBUF   ;TRANSMIT A 2
 013110  017701  167074                 MOV     @CSR,   R1      ;CHECK FOR XMTR READY
 013114  100001                 BPL     .+4             ;BRANCH IF XMTR READY CLEARED
 013116  104001                 HLT+1                   ;TRANSMITTER READY FAILED TO CLEAR
                                                         ;R1 = CONTENTS OF CSR
 013120  013700  001302                 MOV     TIMER,  R0      ;SET UP TIMER
 013124  105305         3$:     DECB    R5              ;SHORT WAIT LOOP
 013126  001376                 BNE     3$
 013130  017701  167054                 MOV     @CSR,   R1      ;SAVE CSR FOR THE RECORD
 013134  000400                 BR      .+2             ;NOP FOR TIMING
```

```
 2898  012136  005300                  DEC     R0              ;TIMER COUNT
 2899  012140  001371                  BNE     3$              ;BRANCH IF MORE TIME
 2900
 2901  012142  022701  100605          CMP     #100605,R1      ;CHECK CSR
 2902  012146  001401                  BEQ     .+4             ;BRANCH IF OK
 2903  012150  104001                  HLT+1                   ;DONE DIDN'T SET OR OTHER CSR ERROR
 2904                                                          ;R1 = CONTENTS OF CSR
 2905  012152  017701  167034          MOV     @RBUF,  R1      ;CHECK RBUF FOR CHAR PRESENT
 2906  012156  100401                  SMI     .+4             ;BRANCH IF CHAR PRESENT
 2907  012160  104001                  HLT+1                   ;CHAR PRESENT MISSING
 2908                                                          ;R1 = CONTENTS OF RBUF
 2909  012162  022701  100001          CMP     #100001,R1      ;CHECK THE DATA
 2910  012166  001401                  BEQ     .+4             ;BRANCH IF OK
 2911  012170  104001                  HLT+1                   ;RECEIVER ERROR
 2912                                                          ;R1 = CNTENTS OF RBUF
 2913  012172  017701  167014          MOV     @RBUF,  R1      ;CHECK RBUF FOR SECOND CHAR
 2914  012176  100401                  BMI     .+4             ;BRANCH IF CHAR PRESENT
 2915  012200  104001                  HLT+1                   ;CHAR PRESENT MISSING
 2916                                                          ;R1 = CONTENTS OF RBUF
 2917  012202  022701  100002          CMP     #100002,R1      ;CHECK THE DATA
 2918  012206  001401                  BEQ     .+4             ;BRANCH IF OK
 2919  012210  104001                  HLT+1                   ;RECEIVER ERROR
 2920                                                          ;R1 = CNTENTS OF RBUF
 2921  012212  017701  166774          MOV     @RBUF,  R1      ;CHECK FOR NO MORE CHARACTERS
 2922  012216  100001                  BPL     .+4             ;BRANCH IF CHAR PRESENT CLEARED
 2923  012220  104001                  HLT+1                   ;CHAR PRESENT NOT CLEAR!
 2924                                                          ;R1 = CONTENTS OF RBUF
 2925  012222  017701  166762          MOV     @CSR,   R1      ;SAVE CSR
 2926  012226  022701  100405          CMP     #100405,R1      ;CHECK CSR
 2927  012232  001401                  BEQ     .+4             ;BRANCH IF OK
 2928  012234  104001                  HLT+1                   ;CSR ERROR
 2929                                                          ;R1 = CONTENTS OF CSR
 2930  012236  005077  166752          CLR     @TCR            ;CLEAR TCR
 2931  012242  005077  166742          CLR     @CSR            ;CLEAR CSR
 2932  012246  104400                  SCOPE
 2933
 2934
 2935
 2936
 2937                                  ;*************************************************************************
 2938                                  ;TEST 54:       TEST THAT RECEIVER ENABLE ON A 0 INHIBITS DONE
 2939                                  ;                 AND CHARACTER PRESENT.
 2940                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-7) E32, E15
 2941                                  ;*************************************************************************
 2942
 2943  012250  012777  000414  166732  TST54:  MOV     #414,   @CSR    ;BIT2 = MAINTENANCE
 2944                                                                  ;BIT3 = CLEAR MOS
 2945                                                                  ;BIT8 = MASTER TRAN SCAN ENB
 2946  012256  032777  000020  166724  10$:    BIT     #BIT4,  @CSR    ;WAIT FOR MOS TO CLEAR
 2947  012264  001374                  BNE     10$
 2948  012266  052777  00000.  166720          BIS     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE0
 2949  012274  017701  166710          1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
 2950  012300  100375                  BPL     1$
 2951  012302  012777  000252  166706          MOV     #252,   @TBUF   ;SEND AN "*"
 2952  012310  013700  001302                  MOV     TIMER,  R0      ;SET UP TIMER
 2953  012314  105305                  2$:     DECB    R5              ;SHORT WAIT LOOP
 2954  012316  001376                  BNE     2$
 2955  012320  017701  166664                  MOV     @CSR,   R1      ;SAVE CSR FOR TYPING
```

```
2954  012324  105701                        TSTB    R1              ;CHECK FOR DONE
2955  012326  100002                        BPL     3$              ;BRANCH IF NOT SET
2956  012330  104001                        HLT+1                   ;DONE SET WHEN RCV ENB CLR
2957                                                                  ;R1=CONTENTS OF CSR
2958  012332  000402                        BR      4$
2959  012334  005300              3$:       DEC     R0              ;TIMER COUNT
2960  012336  001366                        BNE     2$              ;BRANCH IF MORE TIMER
2961
2962  012340  017701  166646        4$:     MOV     @RBUF,  R1      ;CHECK AND SAVE FI/FO
2963  012344  100001                        BPL     .+4             ;BRANCH IF OK
2964  012346  104001                        HLT+1                   ;CHARACTER PRESENT IN FI/FO
2965                                                                  ;R1=DATA FROM FI/FO
2966  012350  005277  166634                INC     @CSR            ;SET RECEIVER ENABLE
2967  012354  017701  166630        5$:     MOV     @CSR,   R1      ;SAVE CSR
2968  012360  105701                        TSTB    R1              ;CHECK FOR DONE
2969  012362  100403                        BMI     6$              ;BRANCH IF OK
2970  012364  105300                        DECB    R0              ;SHORT TIMER
2971  012366  001372                        BNE     5$
2972  012370  104001                        HLT+1                   ;DONE DIDN'T COME UP WHEN RECEIVER ENABLED
2973                                                                  ;UART SHOULD HAVE HELD A CHARACTER
2974                                                                  ;R1 = CONTENTS OF CSR
2975  012372  017701  166614        6$:     MOV     @RBUF,  R1      ;CHECK FOR CHARACTER PRESENT
2976  012376  100401                        BMI     .+4             ;BRANCH IF OK
2977  012400  104001                        HLT+1                   ;CHARACTER PRESENT MISSING
2978                                                                  ;R1 = CONTENTS OF RBUF
2979  012402  005077  166606                CLR     @TCR            ;CLR TRANS CONTROL REG
2980
2981  012406  104400                        SCOPE
2982
2983                                ;*************************************************************************
2984                                ;TEST 55:       TEST THAT HALF DUPLEX (BIT1) DISABLES THE RECEIVER UARTS.
2985                                ;PROBABLE FAULTY LOGIC:  M7285 (D2-4) E32, E17, E22, (D2-2) E5, E1
2986                                ;*************************************************************************
2987
2988  012410  004737  015556        TST55:  JSR     PC,     @#INITC ;INITIALIZE
2989                                                                  ;BIT1 = HALF DUPLEX
2990                                                                  ;BIT2 = MAINTENANCE
2991                                                                  ;BIT3 = CLEAR MOS
2992                                                                  ;BIT8 = MASTER TRAN SCAN ENB
2993                                                                  ;WAIT FOR MOS TO CLEAR
2994                                                                  ;BIT0 = RECEIVER ENABLE
2995  012414  012777  000001  166572        MOV     #BIT0,  @TCR    ;SET XMTR CONTROL BIT, LINE0
2996  012422  017701  166562        1$:     MOV     @CSR,   R1      ;WAIT FOR XMTR READY
2997  012426  100375                        BPL     1$
2998  012430  012777  000252  166560        MOV     #252,   @TBUF   ;SEND AN "*"
2999  012436  013700  001302                MOV     TIMER,  R0      ;SET UP TIMER
3000  012442  105305              2$:       DECB    R5              ;SHORT WAIT LOOP
3001  012444  001376                        BNE     2$
3002  012446  017701  166536                MOV     @CSR,   R1      ;SAVE CSR
3003  012452  105701                        TSTB    R1              ;CHECK FOR DONE
3004  012454  100002                        BPL     3$              ;BRANCH IF NOT SET
3005  012456  104001                        HLT+1                   ;DONE SET WHEN HALF DUPLEX (BIT1) SET
3006                                                                  ;R1=CONTENTS OF CSR
3007  012460  000402                        BR      4$
3008  012462  005300              3$:       DEC     R0              ;TIMER COUNT
3009  012464  001366                        BNE     2$              ;BRANCH IF MORE TIMER
```

```
3010
3011  012466  017701  166520          4$:   MOV     @RBUF,  R1      ;CHECK AND SAVE FI/FO
3012  012472  100001                        BPL     .+4             ;BRANCH IF OK
3013  012474  104001                        HLT+1                   ;CHARACTER PRESENT IN FI/FO
3014                                                                  ;R1=DATA FROM FI/FO
3015  012476  042777  000002  166504        BIC     #BIT1,  @CSR    ;CLEAR HALF DUPLEX BIT
3016  012504  000240                        NOP
3017  012506  000240                        NOP
3018  012510  000240                        NOP
3019  012512  000240                        NOP
3020  012514  017701  166472          5$:   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 OCCURES 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 FI/FO OVERRUN
3121                          ;               THE FI/FO 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  015566          TST60:  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 DIDN'T COME UP
                                                                ;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

                                        TSTB    R1              ;CHECK THAT DONE IS SET
                                        BMI     .+4             ;BRANCH IF OK
                                        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 OVERRUN INTERRUPT
                                ;               DOESN'T OCCUR WHEN THE PROCESSOR IS AT LEVEL 5
                                ;*************************************************************************
013332  012737  000340  177776  T600:   BIC     #340,   @#PS    ;CLEAR PSW
013340  012737  000340  177776          BIS     #340,   @#PS    ;SET PROCESSOR TO LEVEL 5
013346  012737  013464  000304          MOV     #3$,    @RCVVEC ;SET UP RECEIVER INTERRUPT VE:
013354  012737  000340  000310          MOV     #340,   @RCVLVL
013362  012737  100000  177560          BIS     #100000,@#CSR   ;SET STATUS ENABLE
013370  013701  177560                  MOV     @#CSR,  R1      ;SAVE CSR
013374  026127  110605  100000          CMP     #110605,R1      ;CHECK CSR
013402  001401                          BEQ     .+4
013404  104001                          HLT+1                   ;CSR ERROR
                                                                ;R1=CONTENTS OF CSR
013406  000327  177564                  SWAB    @#TBUF          ;SEND LINE #
013412  005737  177564          30$:    TST     @#CSR           ;WAIT FOR TRANSMITTER READY
013416  100375                          BPL     30$
013420  000327  177564                  SWAB    @#TBUF          ;SEND LINE #
013424  012701  177560          31$:    MOV     @#CSR,  R1      ;SAVE CSR
013430  032701  020000                  BIT     #20000, R1      ;WAIT FOR FI/FO OVERRUN
013434  001403                          BEQ     33$
013436  000416                          BR      33$             ;SKIP ISR
013440  012701  177562          32$:    MOV     @#RBUF, R1      ;SAVE CSR
013444  105737  000001                  TSTB    HLT+1           ;INTERRUPT OCCURRED AT LEVEL 5
013450  005737  177562                  TST     @#RBUF          ;"POP" ONE CHARACTER
013454  012716  013466                  MOV     #34$,   (SP)    ;RESET RETURN ADDRESS
013460  000002                          RTI                     ;RETURN
013462  012700  000002          33$:    MOV     #2,     R0      ;SET UP COUNTER - 2 CHARACTERS
013466  012701  177562          34$:    MOV     @#RBUF, R1      ;READ ONE CHARACTER
013472  100401                          BMI     .+4             ;BRANCH IF CHARACTER PRESENT
013474  104001                          HLT+1
013476  032701  070000                  BIT     #70000, R1      ;CHECK ERRORS
013502  001401                          BEQ     .+4
013504  104001                          HLT+1
013506  010102                          MOV     R1,     R2      ;DUP DATA
013510  000302                          SWAB    R2
013512  042702  177760                  BIC     #177760,R2      ;CLR ALL BUT LINE #
013516  120102                          CMPB    R1,     R2      ;CHECK DATA
013520  001401                          BEQ     .+4
013522  104001                          HLT+1
013524  005300                          DEC     R0              ;COUNT CHARACTERS
013526  003005                          BGTR    35$             ;BRANCH IF DONE
013530  105302                  35$:    DECB    R2              ;SHORT WAIT LOOP
013532  001376                          BNE     35$
013534  000754                          BR      34$             ;GO READ ANOTHER CHARACTER
013536  012701  177564          36$:    MOV     @#CSR,  R1      ;SAVE CSR
013542  026127  110605                  CMP     #110605,R1      ;CHECK THAT FI FO OVERRUN CLEARED
013546  001401                          BEQ     .+4             ;BRANCH IF OK
```

```
 3290  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
                                ;*************************************************************************

 013542  042737  000340  177776  T600:   BIC     #340.   @#PS    ;CLEAR PROCESSOR LEVEL
 013550  052737  000200  177776          BIS     #200.   @#PS    ;SET PROCESSOR TO LEVEL 4
 013556  012737  013642  000434          MOV     #42$.   @#OVVEC ;SET UP RECEIVER INTERRUPT VEC
 013564  017701  165420                  MOV     @#CSR,  R1      ;SAVE CSR
 013570  022701  110605                  CMP     #110605,R1      ;CHECK CSR
 013574  001401                          BEQ     .+4
 013576  104001                          HLT+1

 013600  000377  165412                  SWAB    @#TBUF          ;SEND LINE #
 013604  005777  165400          40$:    TST     @#CSR           ;WAIT FOR TRANSMITTER READY
 013610  100375                          BPL     40$
 013612  000377  165400                  SWAB    @#TBUF          ;SEND LINE #
 013616  017701  165366          41$:    MOV     @#CSR,  R1      ;SAVE CSR
 013622  032701  020000                  BIT     #BIT13, R1      ;WAIT FOR FI/FO OVERRUN
 013626  001773                          BEQ     41$
 013630  104001                          HLT+1                   ;INTERUPT DIDN'T OCCURE WHEN OVERRUN SET
                                                                  ;R1 = CONTENTS OF CSR
 013632  052777  000010  165350          BIS     #BIT3,  @#CSR   ;CLEAR MOS
 013640  000464                          BR      45$             ;SKIP TO THE END

 013642  017701  165342          42$:    MOV     @#CSR,  R1      ;SAVE CSR
 013646  022701  130605                  CMP     #130605,R1      ;CHECK CSR
 013652  001401                          BEQ     .+4             ;BRANCH IF OK
 013654  104001                          HLT+1                   ;CSR ERROR
                                                                  ;R1 = CONTENTS OF CSR
 013656  012700  000101                  MOV     #101,   R0      ;SET UP COUNTER - 65 CHARACTERS
 013662  017701  165324          43$:    MOV     @#RBUF, R1      ;READ ONE CHARACTER
 013666  100401                          BMI     .+4             ;BRANCH IF CHARACTER PRESENT
 013670  104001                          HLT+1

 013672  032701  070000                  BIT     #70000. R1      ;CHECK ERRORS
 013676  001401                          BEQ     .+4
 013700  104001                          HLT+1

 013702  010102                          MOV     R1,     R2      ;DUP DATA
 013704  000302                          SWAB    R2
 013706  042702  177760                  BIC     #177760,R2      ;CLR ALL BUT LINE #
 013712  120102                          CMPB    R1,     R2      ;CHECK DATA
 013714  001401                          BEQ     .+4
 013716  104001                          HLT+1

 013720  017701  165264                  MOV     @#CSR,  R1      ;SAVE CSR
 013724  005300                          DEC     R0
 013726  001410                          BEQ     44$
 013730  022700  000100                  CMP     #100.   R0      ;CHECK FOR FIRST CHAR READ
 013734  001752                          BEQ     43$             ;SKIP CSR CHECK ON FIRST CHAR
 013736  022701  110605                  CMP     #110605,R1      ;CHECK CSR
```

```
013742  001401                  BEQ     .+4
013744  104001                  HLT+1

013746  000745                  BR      43$

013750  022701  110405  44$:    CMP     #110405,R1      ;CHECK CSR
013754  001401                  BEQ     .+4             ;BRANCH IF OK
013756  104001                  HLT+1                   ;DONE DIDN'T GO AWAY
                                                        ;OR OTHER CSR ERROR

013760  017701  165226          MOV     @RBUF,  R1      ;READ A CHARACTER
013764  100001                  BPL     .+4             ;BRANCH IF NO CHAR. PRES
013766  104001                  HLT+1                   ;CHAR. PRESENT!

013770  013777  001262  165262  MOV     RCVLVL, @RCVVEC
013776  012777  000004  165216  MOV     #1OT,   @RCVLVL
014004  012716  014012          MOV     #45$,   (SP)    ;RESET RETURN ADDRESS ON STACK
014010  000002                  RTI                     ;RESTORE PSW
014012  005077  165176  45$:    CLR     @TCR
014016  005077  165166          CLR     @CSR
014022  042737  000340  177776  BIC     #340,   @#PS    ;LOWER PROCESSOR STATUS

014030  104400                  SCOPE


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

014032  012777  000414  165150  TST61:  MOV     #414,   @CSR    ;BIT2 = MAINTENANCE
                                                        ;BIT3 = CLEAR MOS
                                                        ;BIT8 = TRANS SCAN ENABLE
014040  032777  000020  165142  10$:    BIT     #BIT4,  @CSR    ;WAIT FOR MOS TO CLEAR
014046  001374                  BNE     10$
014050  005001                  CLR     R1              ;SET UP LINE COUNTER
014052  012777  000001  165134  MOV     #1,     @TCR    ;TRANS CONTROL, LINE 0
014060  017702  165124  LOP61:  MOV     @CSR,   R2      ;WAIT FOR TRANS READY
014064  100375                  BPL     LOP61
014066  012777  000001  165122  MOV     #1,     @TBUF   ;SEND #1
014074  017702  165110  2$:     MOV     @CSR,   R2      ;WAIT FOR TRANS READY
014100  100375                  BPL     2$
014102  012777  000002  165106  MOV     #2,     @TBUF   ;SEND #2
014110  013700  001302          MOV     TIMER,  R0      ;SET UP TIMER
014114  105305          3$:     DECB    R5              ;SHORT TIME LOOP
014116  001376                  BNE     3$
014120  017702  165064          MOV     @CSR,   R2      ;SAVE CSR FOR THE RECORD
014124  000400                  BR      .+2             ;NOP FOR TIMING
014126  005300                  DEC     R0              ;TIMER COUNT
014130  001371                  BNE     3$
014132  005277  165052          INC     @CSR            ;SET RECEIVER ENABLE
014136  017702  165046          MOV     @CSR,   R2      ;SAVE CSR
014142  032702  000001          BIT     #BIT0,  R2      ;CHECK RECEIVER ENABLE
014146  001001                  BNE     .+4             ;BRANCH IF OK
014150  104002                  HLT+2                   ;RECEIVER ENABLE FAILED TO SET
                                                        ;R1 = LINE #
```

```
                                                        ;R2 = CONTENTS OF CSR
014152  105777  165032          4$:   TSTB    @CSR      ;WAIT FOR DONE
014156  100375                  BPL   4$
014160  017702  165026          MOV   @RBUF, R2         ;READ FI/FO
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 OVERRRUN
014174  001001                  BNE   .+4               ;BRANCH IF OK
014176  104002                  HLT+2                   ;UART OVERRUN MISSING
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
014200  122702  000002          CMPB  #2,     R2        ;CHECK THE DATA
014204  001401                  BEQ   .+4               ;BRANCH IF OK
014206  104002                  HLT+2                   ;DATA ERROR - 3RD CHAR OVERRUNS 2ND
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
014210  010203                  MOV   R2,     R3        ;DUP DATA
014212  000303                  SWAB  R3
014214  042703  177700          BIC   #177700,R3        ;MASK ALL BIT LINE #, ERROR BITS
014220  020103                  CMP   R1,     R3        ;CHECK LINE #, ERRORS
014222  001401                  BEQ   .+4               ;BRANCH IF OK
014224  104002                  HLT+2                   ;LINE # OR OTHER RBUF ERROR
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
014226  017702  164760          7$:   MOV   @RBUF, R2   ;CHECK FOR MORE DATA
014232  100002                  BPL   8$                ;BRANCH IF OK
014234  104002                  HLT+2                   ;EXTRA DATA IN FI/FO!
                                                        ;R1 = LINE #
                                                        ;R2 = CONTENTS OF RBUF
014236  000773                  BR    7$

014240  005377  164744          8$:   DEC   @CSR
014244  017702  164740          MOV   @CSR, R2          ;SAVE CSR
014250  022702  100404          CMP   #100404,R2        ;CHECK CSR
014254  001401                  BEQ   .+4               ;BRANCH IF OK
014256  104002                  HLT+2                   ;CSR ERROR
                                                        ;R1 = LINE NUMBER
                                                        ;R2 = CONTENTS OF CSR
014260  005201                  INC   R1                ;COUNT LINES
014262  006377  164726          ASL   @TCR              ;GO TO NEXT LINE
014266  103274                  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, E2, E19, E35
                        ;******************************************************************************

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

```
                                                MOV     @BCSR,  R1      ;CHECK AND SAVE BCSR
                                                CMP     #177777,R1      ;CHECK THAT ALL THE BITS ARE SET
                                                BEQ     .+4             ;BRANCH IF OK
                                                HLT+1                   ;BIT(S) OF BCSR FAILED TO SET

                                                CLR     @BCSR           ;CLEAR BCSR
                                                MOV     @BCSR,  R1      ;CHECK THAT IT CLEARED AND SAVE
                                                BEQ     .+4             ;BRANCH IF CLR
                                                HLT+1                   ;BIT(S) OF BCSR FAILED TO CLEAR

                                                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, E35
                                ;*****************************************************************************

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
014350  012777  000001  164636        MOV     #1,     @BCSR   ;SEND BREAKS, LINE 0
014356  013700  001302                MOV     TIMER,  R0      ;SET UP TIMER
014362  105305                1$:     DECB    R5              ;SHORT WAIT LOOP
014364  001376                        BNE     1$
014366  017701  164616                MOV     @CSR,   R1      ;SAVE CSR
014372  105701                        TSTB    R1              ;CHECK FOR DONE
014374  100404                        BMI     2$              ;BRANCH WHEN DONE
014376  005200                        INC     R0              ;WAIT A WHILE
014400  001370                        BNE     1$
014402  104001                        HLT+1                   ;DONE NEVER CAME UP
                                                                ;R1=CONTENTS OF CSR
014404  000430                        BR      4$
014406  022701  002205        2$:     CMP     #2205,  R1      ;CHECK REST OF CSR
014412  001401                        BEQ     .+4             ;BRANCH IF OK
014414  104001                        HLT+1                   ;CSR ERROR
                                                                ;R1=CONTENTS OF CSR

014416  017701  164570                MOV     @RBUF,  R1      ;GET DATA FROM FIFO
014422  100401                        BMI     .+4             ;BRANCH IF OK
014424  104001                        HLT+1                   ;CHARACTER PRESENT NOT UP
                                                                ;R1=CONTENTS OF RBUF

014426  032701  020000                BIT     #020000,R1      ;CHECK FOR FRAMING ERROR
014432  001001                        BNE     .+4             ;BRANCH IF OK
014434  104001                        HLT+1                   ;FRAMING ERROR NOT UP
                                                                ;R1=CONTENTS OF RBUF
014436  133737  001310  001276        BITB    DJPAR,  PARITY  ;CHECK ODD PARITY FLAG
014444  001404                        BEQ     3$              ;BRANCH IF NOT
014446  032701  010000                BIT     #010000,R1      ;CHECK PARITY ERROR
014452  001005                        BNE     4$              ;
```

```
 3514  014454  104001                          HLT+1                   ;ODD PARITY SHOULD CAUSE PARITY ERROR
 3515                                                                  ;R1=CONTENTS OF RBUF
 3516
 3517  014456  032701  010000        3$:       BIT     #010000,R1      ;CHECK PARITY ERROR
 3518  014462  001401                          BEQ     .+4             ;BRANCH IF OK
 3519  014464  104001                          HLT+1                   ;EVEN PARITY OR NO PARITY
 3520                                                                  ;SHOULDN'T CAUSE PARITY ERROR
 3521                                                                  ;R1=CONTENTS OF RBUF
 3522  014466  032701  040000        4$:       BIT     #040000,R1      ;CHECK UART OVERRUN
 3523  014472  001401                          BEQ     .+4             ;BRANCH IF OK
 3524  014474  104001                          HLT+1                   ;UART OVERRUN SET!!
 3525                                                                  ;R1=CONTENTS OF RBUF
 3526  014476  032701  007400                  BIT     #007400,R1      ;CHECK LINE #
 3527  014502  001401                          BEQ     .+4             ;BRANCH IF OK
 3528  014504  104001                          HLT+1                   ;WRONG LINE# IN FIFO
 3529                                                                  ;R1=CONTENTS OF RBUF
 3530  014506  105701                          TSTB    R1              ;CHECK DATA
 3531  014510  001401                          BEQ     .+4             ;BRANCH IF OK
 3532  014512  104001                          HLT+1                   ;WRONG DATA RECEIVED
 3533                                                                  ;R1=CONTENTS OF RBUF
 3534
 3535  014514  017701  164472                  MOV     @RBUF,  R1      ;READ FIFO AGAIN
 3536  014520  100001                          BPL     .+4             ;BRANCH IF OK
 3537  014522  104001                          HLT+1                   ;EXTRA CHAR IN FIFO
 3538                                                                  ;R1 = CONTENTS OF RBUF
 3539  014524  005077  164464                  CLR     @BCSR           ;CLEAR BREAK CONTROL REG
 3540  014530  105305                  5$:     DECB    R5              ;SHORT WAIT LOOP-REGISTER A MARK
 3541  014532  001376                          BNE     5$
 3542
 3543  014534  104400                          SCOPE
 3544
 3545
 3546                                  ;*************************************************************************
 3547                                  ;TEST 64:       TEST THAT EACH LINE CAN TRANSMIT AND RECEIVE A BREAK
 3548                                  ;               ALSO CHECKS FRAMING ERROR(RBUF BIT13)
 3549                                  ;               ALSO CHECKS PARITY ERROR(RBUF BIT12)
 3550                                  ;                 IF ODD PARITY IS SELECTED
 3551                                  ;PROBABLE FAULTY LOGIC:  M7285 (D2-2) E5, E1, (D2-3) E16, E3, E19, E35
 3552                                  ;*************************************************************************
 3553
 3554  014536  004737  015536  TST64:  JSR     PC,     I#INITA ;INITIALIZE
 3555                                                          ;BIT2 = MAINTENANCE
 3556                                                          ;BIT3 = CLEAR MOS
 3557                                                          ;BIT10= R/W BCSR
 3558                                                          ;WAIT FOR MOS TO CLEAR
 3559                                                          ;BIT0 = RECEIVER ENABLE
 3560  014542  005001                          CLR     R1              ;SET UP LINE COUNTER
 3561  014544  012704  000001                  MOV     #1,     R4      ;SET UP LINE MARKER
 3562  014550  013700  001302  LOP64:  MOV     TIMER,  R0              ;SET UP TIMER
 3563  014554  050477  164434                  BIS     R4,     @BCSR   ;SET BREAK CONTROL BIT, LINE # IN R1
 3564  014560  105305          1$:     DECB    R5                      ;SHORT WAIT LOOP
 3565  014562  001376                          BNE     1$
 3566  014564  017702  164420                  MOV     @CSR,   R2      ;SAVE CSR
 3567  014570  105702                          TSTB    R2              ;CHECK FOR DONE
 3568  014572  100404                          BMI     2$              ;BRANCH WHEN DONE
 3569  014574  005200                          INC     R0              ;WAIT A WHILE
```

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

014602  000433                          BR      4$
014604  022702  002205          2$:     CMP     #2205,  R2      ;CHECK REST OF CSR
014610  001401                          BEQ     .+4             ;BRANCH IF OK
014612  104002                          HLT+2                   ;CSR ERROR
                                                                ;R1 = LINE #
                                                                ;R2 = CONTENTS OF CSR

014614  017702  164372                  MOV     @RBUF,  R2      ;GET DATA FROM FI/FO
014620  100401                          BMI     .+4             ;BRANCH IF OK
014622  104002                          HLT+2                   ;CHARACTER PRESENT NOT UP
                                                                ;R1 = LINE #
                                                                ;R2 = CONTENTS OF RBUF

014624  032702  020000                  BIT     #020000,R2      ;CHECK FOR FRAMING ERROR
014630  001001                          BNE     .+4             ;BRANCH IF OK
014632  104002                          HLT+2                   ;FRAMING ERROR NOT UP
                                                                ;R1 = LINE #
                                                                ;R2 = CONTENTS OF RBUF

014634  010103                          MOV     R1,     R3      ;GET LINE #
014636  006203                          ASR     R3              ;/4
014640  006203                          ASR     R3
014642  133763  001310  001276          BITB    DJPAR,PARITY(3) ;CHECK ODD PARITY FLAG
014650  001404                          BEQ     3$              ;BRANCH IF NOT
014652  032702  010000                  BIT     #010000,R2      ;CHECK PARITY ERROR
014656  001005                          BNE     4$              ;
014660  104002                          HLT+2                   ;ODD PARITY SHOULD CAUSE PARITY ERROR
                                                                ;R1 = LINE #
                                                                ;R2 = CONTENTS OF RBUF

014662  032702  010000          3$:     BIT     #010000,R2      ;CHECK PARITY ERROR
014666  001401                          BEQ     4$              ;BRANCH IF OK
014670  104002                          HLT+2                   ;EVEN PARITY OR NO PARITY
                                                                ;SHOULDN'T CAUSE PARITY ERROR
                                                                ;R1 = LINE #
                                                                ;R2 = CONTENTS OF RBUF
014672  032702  040000          4$:     BIT     #040000,R2      ;CHECK UART OVERRUN
014676  001401                          BEQ     .+4             ;BRANCH IF OK
014700  104002                          HLT+2                   ;UART OVERRUN SET!!
                                                                ;R1 = LINE #
                                                                ;R2 = CONTENTS OF RBUF

014702  010203                          MOV     R2,     R3      ;DUP DATA
014704  000303                          SWAB    R3              ;LINE # IN LOW BYTE
014706  042703  177760                  BIC     #177760,R3      ;MASK ALL BUT LINE #
014712  020103                          CMP     R1,     R3      ;CHECK FOR RIGHT LINE #
014714  001401                          BEQ     .+4             ;BRANCH IF OK
014716  104002                          HLT+2                   ;WRONG LINE# IN FI/FO
                                                                ;R1 = LINE #
                                                                ;R2 = CONTENTS OF RBUF


014720  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 BERECEIVING
;         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
```

```
                                        SCOPE

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

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,@CSR        ;CLEAR MOS
015142  012777  010505  164040  MOV     #010505,@CSR    ;BIT0 = RECEIVE ENABLE
                                                        ;BIT2 = MAINTENANCE
                                                        ;BIT6 = RECEIVER INTERRUPT ENB
                                                        ;BIT8 = TRANS SCAN ENABLE
                                                        ;BIT12= STATUS ENABLE
015150  032777  000020  164032  10$:     BIT     #BIT4,  @CSR    ;WAIT FOR MOS TO CLEAR
015156  001374                  BNE     10$
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     @CSR            ;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     @CSR,   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
```

```
                                                                        ;R1=LINE #
                                                                        ;R2=CONTENTS OF RBUF
                                                                        ;R3=EXPECTED DATA
015304  020203                  11$:   CMP     R2,     R3               ;CHECK THE DATA
015306  001401                         BEQ     .+4                      ;BRANCH IF OK
015310  104003                         HLT+3                            ;DATA ERROR
                                                                        ;R1=LINE #
                                                                        ;R2=CONTENTS OF RBUF
                                                                        ;R3=EXPECTED DATA
015312  105203                         INCB    R3                       ;UPDATE EXPECTED DATA
015314  147703  163766                 BICB    @DJLEN, R3               ;MASK CHARACTER LENGTH
015320  001403                         BEQ     12$                      ;BRANCH OUT IF DATA=0
015322  017702  163664                 MOV     @RBUF,  R2               ;READ MORE DATA
015326  100766                         BMI     11$                      ;BRANCH IF MORE
015330  000002                 12$:    RTI                              ;RETURN

015332  162737  000004  001306 END66:  SUB     #4,     DJLEN            ;RESET CHAR LENGTH POINTER
015340  013777  001222  163652         MOV     RCVLVL, @RCVVEC          ;RESTORE RECEIVER INT. VEC
015346  012777  000004  163646         MOV     #IOT,   @RCVLVL
015354  005077  163634                 CLR     @TCR                     ;CLEAR TCR
015360  005077  163624                 CLR     @CSR                     ;CLEAR CSR
015364  104400                         SCOPE

015366  012737  000020  015772         MOV     #20,    TIMES
015374  023737  001304  001234         CMP     DJUUT,UNITS              ;CHECK FOR LAST UNIT
015402  002002                         BGE     DONE                     ;BRANCH IF LAST UNIT
015404  000137  002302                 JMP     RESTAR                   ;JUMP IF NOT

015410                         DONE:
015410  004737  017640                 JSR     PC,     KBDINT
015414  062737  000001  001206         ADD     #1,PCNT+2                ;ADD 1 TO THE PASS COUNT
015422  005537  001204                 ADC     PCNT                     ;MAKE IT DOUBLE PREC.
015426  032777  002000  163662         BIT     #SW10,@SWR               ;RING THE BELL?
015434  001004                         BNE     1$                       ;NO!
015436  000004  000007                 TYPE    ,BELL                    ;RING THE BELL
015442  000004  000177                 TYPE    ,177                     ;TYPE A FILLER FOR 11/05
015446  005046                 1$:     CLR     -(6)                     ;CLEAR TRACE TRAP
015450  032777  010000  163640         BIT     #SW12,@SWR               ;RUN WITH TRT?
015456  001010                         BNE     2$                       ;SKIP T BIT
015460  005137  015532                 COM     .TBIT                    ;COMPLIMENT FLAG
015464  100005                         BPL     2$                       ;SKIP IF PLUS
015466  052716  000020                 BIS     #20,(6)                  ;SET TRACE TRAP
015472  012746  015526                 MOV     #3$,-(6)                 ;JUMP TO START OF TEST
015476  000002                         RTI                              ;RETURN
015500  012746  015506         2$:     MOV     #4$,-(6)                 ;JUMP TO START OF TEST
015504  000002                         RTI                              ;RETURN
015506  013700  000042         4$:     MOV     @#42,R0                  ;GET MONITOR ADDRESS
015512  001405                         BEQ     3$                       ;IF NONE
015514  000005                         RESET                            ;RESET AND
        015516                         $ENDAD = .
015516  004710                         JSR     7,(0)                    ;GO TO MONITOR
015520  000240                         NOP                              ;SAVE ROOM
015522  000240                         NOP                              ;FOR
015524  000240                         NOP                              ;ACT11
015526  000137  002302         3$:     JMP     RESTAR                   ;RETURN
```

```
 3794  015532  000000                    .TBIT:  0                       ;T BIT FLAG
 3795
 3796  015534  000002                    YESRT:  RTI                     ;RETURN FROM TRACE TRAP
 3797                                    ;
 3798                                    ;   INITIALIZATION ROUTINE
 3799                                    ;
 3800                                    ;       DEVICE CSR REGISTER
 3801                                    ;
 3802                                    ;ON FAILURE:    REGISTER 0 CONTAINS ERROR ADDRESS
 3803                                    ;
 3804                                    ;SET:
 3805                                    ;BIT01 = HALF DUPLEX
 3806                                    ;BIT02 = MAINTENANCE
 3807                                    ;BIT03 = CLEAR MOS
 3808                                    ;BIT06 = RECEIVER INTERRUPT ENABLE
 3809                                    ;BIT08 = MASTER XMTR SCAN ENB
 3810                                    ;BIT10 = R/W BCSR
 3811                                    ;
 3812                                    ;WAIT FOR MOS TO CLEAR
 3813                                    ;SET:
 3814                                    ;BIT00 = RECEIVER ENABLE
 3815                                    ;
 3816                                                            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
```

```
3850                                    ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
3851                                    ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
3852
3853  015632  004737  017640            TRAP$:  JSR     PC,     KBDINT
3854  015636  032777  000400  163452            BIT     #SW8,@SWR       ;LOOP ON SPEC. TEST?
3855  015644  001404                    1$:     BEQ     1$              ;NO LOOP ON SPEC. TEST
3856  015646  127737  163444  001200            CMPB    @SWR,ICNT       ;ON RIGHT TEST?   *SW7-0*
3857  015654  001434                            BEQ     OVER$           ;NOT RIGHT TEST
3858  015656  032777  040000  163432  1$:       BIT     #SW14,@SWR      ;LOOP ON TEST?
3859  015664  001026                            BNE     KIT$            ;LOOP ON TEST IS SET
3860  015666  032777  004000  163422            BIT     #SW11,@SWR      ;KILL ITERATIONS
3861  015674  001012                            BNE     SVLAD$          ;YES - KILL ITERATIONS
3862  015676  105737  001201                    TSTB    ICNT+1          ;FIRST ONE?
3863  015702  001404                            BEQ     2$              ;BRANCH IF FIRST
3864  015704  123737  015772  001201            CMPB    TIMES,ICNT+1    ;DONE?
3865  015712  001013                            BNE     KIT$            ;BRANCH IF NOT
3866  015714  112737  000001  001201  2$:       MOVB    #1,ICNT+1       ;FIRST ITERATION
3867  015722  105237  001200            SVLAD$: INCB    ICNT            ;COUNT TEST NUMBERS
3868  015726  011637  015770                    MOV     (6),LAD         ;SAVE LOOP ADDRESS
3869  015732  013737  001200  001320            MOV     ICNT,@#DISPLAY  ;DISPLAY TEST NO. AND ITERATION COUNT
3870  015740  000002                            RTI                     ;RETURN
3871
3872  015742  105237  001201            KIT$:   INCB    ICNT+1          ;INC THE ITERATION COUNT
3873  015746  013737  001200  001320    OVER$:  MOV     ICNT,@#DISPLAY  ;SET UP DISPLAY
3874  015754  005737  015770                    TST     LAD             ;FIRST ONE?
3875  015760  001760                            BEQ     SVLAD$          ;YES
3876  015762  013716  015770                    MOV     LAD,(6)         ;FUDGE RETURN ADDRESS
3877  015766  000002                            RTI                     ;FIXES PS
3878
3879  015770  000000            LAD:    0               ;LOOP ADDRESS
3880  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

016616  016737  000001  016510  READIN: MOV     #1,INHRE
016622  016364                          JSR     PC,     READ$   ;GO READ TTY UNTIL CR
016626  005004                          CLR     INHRE
016630  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
016632  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
016634  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
016636  011401                          MOV     (R4)+,  R1
016640  012703  000020  020050          MOV     #' ,R3
016644  016702  016510                  MOV     INPUT,R2
016650  122712  000120                  CMPB    #'P,(R2)        ;CHECK FOR "P"
016654  001403                          BEQ     3$
016656  005001                          CLR     (R1)
016660  112203                  1$:     MOVB    (R2)+,R3
016662  122703  000015                  CMPB    R3,#15
016666  001403                          BEQ     3$
016670  162703  000060                  SUB     #60,R3
016674  103003  177770                  BCC     4$
016700  000167  000006                  JMP     @#177770,R3
016704                                  BNE     3$              ;BRANCH IF BAD DATA
016704  005211                          INC     (R1)
016706  006311                          ASL     (R1)
016710  006311                          ASL     (R1)
016712  006311                          ASL     (R1)
016714  060311                          ADD     R3,(R1)
016716  000765                          BR      1$
016720  020050                  3$:     CMP     CNT,    #' 
016724  001002                          BNE     2$
016726  005367  020050                  CLR     CNT
016732  000207                  2$:     RTS     PC              ;MAKE SURE Z-BIT IS CLR
016734  013737  177776  016362  4$:     MOV     @#PS,   PSTEMP  ;SAVE CONDITION CODES
016742  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
016744  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
016746  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
016750  013706  016362  177776          MOV     PSTEMP, @#PS    ;RESTORE CONDITION CODES
016756  000205                          RTS     R5

016362  000000          PSTEMP: 0               ;TEMPORARY STORAGE FOR PS

016364  010346          READ$:  MOV     R3,-(6)         ;SAVE R3
016366  012703  016510  1$:     MOV     #INPUT,R3       ;GET ADDRESS
016372  026703  016532  2$:     CMP     #INPUT+20,R3    ;BUFFER FULL?
016376  001411                  BEQ     4$              ;YES - TYPE "?"
016400  105737  177560          TSTB    @#177560        ;WAIT FOR
016404  100375                  BPL     .-4             ;A CHARACTER
016406  113713  177562          MOVB    @#177562,(3)    ;GET CHARACTER
016412  142713  000200          BICB    #200,(3)        ;GET RID OF JUNK
016416  122713  000177          CMPB    #177,(3)        ;IS IT A RUBOUT
016422  001403                  BEQ     4$              ;SKIP IF NOT
016424  122713  000025          CMPB    #25,(3)
016430  001006                  BNE     3$
016432                  4$:
016432  000004  016436          TYPE    .+2             ;.ASCIZ "?"<15><12>"= "
016436  022603                  CMP     (3),R3          ;CMP THE BUFFER AND LOOP
016440  001355                  BNE     2$
016442  113713  017234          MOVB    (3),.TYPE       ;SET UP FOR TYPING
```

```
016452  000004  017334                  TYPE    ;;TYPE          ;ECHO IT
016456  122703  000015                  CMPB    #15.,(3)+       ;CHECK FOR RETURN
016462  001343                          BNE     2$              ;LOOP IF NOT RETURN
016464  005737  016510                  TST     INHRE
016470  001401                          BEQ     5$
016472  000402                          BR      6$
016474  105063  177777          5$:     CLRB    -1(3)           ;ZAP RETURN (THE 15)
016500  000004  000012          6$:     TYPE    ,12             ;TYPE A LINE FEED
016504  012603                          MOV     (6)+,R3         ;RESTORE R3
016506  000207                          RTS     PC              ;RETURN

016510  000000          INHRE:  0
016512  001020          INPUT:  .BLKW   20              ;TTY 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.

016556  012737  170101  016720  BITYP$: MOV     #170101..PR     ;SET BIT FLAG ANS 16. CHARACTER COUNT
016560  000411          BR      .PTIT           ;NOW TYPE IT IN BIT FORM
016562  112737  000001  016720  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
016570  000403          BR      .+6             ;SKIP
016572  005037  016720  PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
016576  112737  177772  016721          MOVB    #-6,..PR+1      ;SET COUNT
016604  010446          .PTIT:  MOV     R4,-(6)         ;SAVE R4
016606  012704  016722          MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
016612  105014          CLRB    (4)             ;CLEAR FIRST BYTE
016614  000411          BR      .PRF            ;ROTATE FIRST BIT
016616  105014          .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
016620  032737  000100  016720          BIT     #100,..PR       ;BIT TYPING MODE?
016626  001004          BNE     .PRF            ;YES - SKIP 2 ROTATES
016630  006105          ROL     TTY             ;ROTATE BIT INTO C
016632  106114          ROLB    (4)             ;PACK IT
016634  006105          ROL     TTY             ;ROTATE BIT INTO C
016636  106114          ROLB    (4)             ;PACK IT
016640  006105          .PRF:   ROL     TTY             ;ROTATE BIT INTO C
016642  106114          ROLB    (4)             ;PACK IT
016644  105714          TSTB    (4)             ;IS IT ZERO?
016646  001402          BEQ     .+6             ;SKIP INC
016650  105237  016720          INCB    .PR             ;SET FILL SWITCH
016654  105737  016720          TSTB    .PR             ;CHECK FILL SWITCH
016660  001402          BEQ     .+6             ;SKIP BITSET
016662  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
016666  105237  016721          INCB    .PR+1           ;INC COUNT
016672  001351          BNE     .PRL            ;REPEAT
016674  022704  016722          CMP     #.PR+2,R4       ;EMPTY BUFFER?
016700  001002          BNE     .+6             ;SKIP IF NOT
016702  112724  000060          MOVB    #'0,(4)+        ;LOAD 1 ZERO
016706  105014          CLRB    (4)             ;NULL TERMINATOR
016710  004737  000004  016722          TYPE    ,.PR+2          ;TYPE IT
016714  012604          MOV     (6)+,R4         ;RESTORE R4
016716  000207          RTS     PC              ;RETURN
016720  000012          .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
016744  012777  017072  000126  PDOWN$: MOV     #ILLUP,@PUVEC$          ;SET FOR FAST UP
016752  012777  000340  000122          MOV     #340,@PUVEC$+2          ;PRIO:7
016760  010046                          MOV     R0,-(6)                 ;PUSH R0 ON STACK
016762  010146                          MOV     R1,-(6)                 ;PUSH R1 ON STACK
016764  010246                          MOV     R2,-(6)                 ;PUSH R2 ON STACK
016766  010346                          MOV     R3,-(6)                 ;PUSH R3 ON STACK
016770  010446                          MOV     R4,-(6)                 ;PUSH R4 ON STACK
016772  010546                          MOV     R5,-(6)                 ;PUSH R5 ON STACK
016774  010637  017076                  MOV     SP,.SAVR6               ;SAVE SP
017000  012777  017010  000072          MOV     #PUP$,@PUVEC$           ;SET UP VECTOR
017006  000000                          HALT                            ;WAIT FOR PF

017010  013706  017076          PUP$:   MOV     .SAVR6,SP               ;GET SP
017014  005001                          CLR     R1                      ;WAIT LOOP FOR THE TTY
017016  005201                  1$:     INC     R1                      ;WAIT FOR THE INC
017020  001376                          BNE     1$                      ;OF  WORD
017022  012605                          MOV     (6)+,R5                 ;POP STACK INTO R5
017024  012604                          MOV     (6)+,R4                 ;POP STACK INTO R4
017026  012603                          MOV     (6)+,R3                 ;POP STACK INTO R3
017030  012602                          MOV     (6)+,R2                 ;POP STACK INTO R2
017032  012601                          MOV     (6)+,R1                 ;POP STACK INTO R1
017034  012600                          MOV     (6)+,R0                 ;POP STACK INTO R0
017036  012737  016744  000024          MOV     #PDOWN$,@#24            ;SET UP THE POWER DOWN VECTOR
017044  012737  000340  000026          MOV     #340,@#26               ;PRIO:7
017052  000004  017056                  TYPE    ,.+2                    ;.ASCIZ <15><12>"POWER"
017066  000137  002302                  JMP     RESTAR                  ;JMP TO USER ADDRESS

017072  000000                  ILLUP:  HALT                            ;THE POWER UP SEQUENCE WAS STARTED
017074  000776                          BR      .-2                     ; BEFORE THE POWER DOWN WAS COMPLETE

017076  000000                  .SAVR6: 0                               ;PUT THE SP HERE
017100  000024  000026          PUVEC$: 24,26                           ;POWER UP VECTOR
```

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

017104  022716  001000          IOTRAP: CMP     #1000,  (SP)    ;CHECK RETURN ADDRESS FOR FAULSE TRAP
017110  002407                          BLT     IOT$            ;BRANCH IF "TYPE" COMMAND INTENDED
017112  162716  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 INTERUPT 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$:   MOV     TTY,-(6)        ;SAVE TTY
017132  017605  000002                  MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
017136  032705  177400                  BIT     #177400,TTY     ;IS IT A TYPEM?
017142  001004                          BNE     1$              ;NO
017144  010537  017234                  MOV     TTY,.TYPE       ;GET THE CHARACTER
017150  012705  017234                  MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
017154  105715                  1$:     TSTB    (TTY)           ;TERMINATOR?
017156  001406                          BEQ     2$              ;GET OUT IF SO
017160  112537  177566                  MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
017164  105737  177564                  TSTB    @#177564        ;IS THE PRINTER READY
017170  100376                          BPL     .-4             ;WAIT UNTIL IT IS
017172  000770                          BR      1$              ;GET THE NEXT CHARACTER
017174  017646  000002          2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
017200  062766  000002  000004          ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
017206  022666  000002                  CMP     (6)+,2(6)       ;IS IT .+2?
017212  001006                          BNE     3$              ;NO
017214  062705  000002                  ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
017220  042705  000001                  BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
017224  010566  000002                  MOV     TTY,2(6)        ;RESTORE ADDRESS
017230  012605                  3$:     MOV     (6)+,TTY        ;RESTORE TTY
017232  000002                          RTI                     ;RETURN
017234  000000                  .TYPE:  0                       ;CHARACTER TYPE LOCATION

017236  005015      000         RETURN: .ASCIZ  <15><12>
017241      040  000040         SPACE:  .ASCIZ  " "
017244  005015  044506  051522 MSGADR: .ASCIZ  <15><12>"FIRST DJ11 ADDRESS:  "
017252  020124  045104  030461
017260  040440  042104  042522
017266  051523  020072  000040
017274  005015  042526  052103 MSGVEC: .ASCIZ  <15><12>"VECTOR ADDRESS:  "
017302  051117  040440  042104
```

```
 4128  017310  042522  051523  020072
 4129  017316  000040
 4130  017320  005015  047516  020056  MSGNUM: .ASCIZ  <15><12>"NO. OF DJ11'S:  "
 4131  017326  043117  042040  030512
 4132  017334  023461  035123  020040
 4133  017342     000
 4134  017343     015  051412  040524  MSGCON: .ASCIZ  <15><12>"STANDARD CONFIGURATION"  "
 4135  017350  042116  051101  020104
 4136  017356  047503  043116  043511
 4137  017364  051105  052101  047511
 4138  017372  037516  020040     000
 4139  017377     015  046012  047111  MSGLIN: .ASCIZ  <15> 12 "LINES "
 4140  017404  051506  000040
                                      MSGDAS: .ASCIZ  " - "
 4141  017410  026440  000040
 4142  017414  005015  044103  051101  MSGLEN: .ASCIZ  <15><12>"CHAR LENGTH:  "
 4143  017420  046040  047106  052107
 4144  017436  035116  020040     000
 4145  017436     015  050012  051101  MSGPAR: .ASCIZ  <15><12>"PARITY(NO, ODD, EVEN):  "
 4146  017446  052111  024131  047516
 4147  017450  020054  042117  026104
 4148  017456  042440  042526  024516
 4149  017464  020072  000040
 4150                                  .EVEN
 4151
 4152                                  ;SUBROUTINE TO AUTOMATICLY DETERMINE THE NUMBER OF DJ11'S ON THE SYSTEM
 4153                                  ;       AND WHERE THEIR INTERUPT VECTORS ARE.
 4154                                  ;THIS ROUTINE IS ONLY USED IF LOC 72 IS NOT ZERO, AS WHEN THE PROGRAM IS
 4155                                  ;       BEING RUN UNDER ACT11 OR DDP MONITOR CONTROL.
 4156                                  ;
 4157                                  ;NOTE:  SOME OF THE LOGIC MUST BE FUNCTIONAL OR THIS ROUTINE WILL BOMB!
 4158
 4159  017470  012700  160000         AUTO:   MOV     #160000,R0      ;START AT NO RESPONSE BASE DJ11 -10 B
 4160  017474  012701  000001                 MOV     #1.,    R1      ;COUNTER OF NON RESPONSE IN DJ11 IS DONE
 4161  017500  012737  000002  000006         MOV     #8$,    @#4     ;$$ WHEN TIME-OUT
 4162  017506  005001                 9$:     CLR     R1              ;SET UP COUNTER
 4163  017510  000261                 10$:    SEC                     ;SET CARRY
 4164  017512  005710                         TST     (R0)            ;CHECK FOR ANY DJ11'S
 4165  017514  103404                         BCS     8$              ;BRANCH IF IT TIMED OUT
 4166  017516  062700  000010                 ADD     #10.,   R0      ;POINT TO NEXT DJ11 ADDRESS
 4167  017522  005201                         INC     R1              ;COUNT DJ11'S
 4168  017524  000771                         BR      10$             ;LOOK FOR MORE
 4169
 4170  017526  005302                 8$:     DEC     R2              ;COUNT DOWN DEVICES
 4171  017530  100405                         BMI     7$              ;BRANCH IF DONE
 4172  017532  062700  000010                 ADD     #10.,   R0      ;POINT TO FIRST DJ11
 4173  017536  010037  001230                 MOV     R0,     @#DEVADR ;SAVE FIRST DJ11 ADDRESS
 4174  017542  000761                         BR      9$              ;GO COUNT DJ11'S
 4175
 4176  017544  005037  000006         7$:     CLR     @#6             ;RESTORE TIME-OUT CATCHER
 4177  017550  010137  001234                 MOV     R1,     @#UNITS ;SAVE COUNT
 4178  017554  001003                         BNE     3$              ;BRANCH IF NOT ZERO
 4179  017556  104000                         HALT                    ;REPORT THAT NO DJ11'S WERE FOUND
 4180  017560  000137  015410                 JMP     DONE            ;EXIT THIS PROGRAM
 4181
 4182                                  ;ROUTINE TO DETERMINE VECTOR ADDRESSES
 4183
 4184  017564  013746  000020         3$:     MOV     @#20,   -(SP)   ;SAVE IOT VECTOR ON THE STACK
```

```
017570  012737  017624  000020          MOV     #4$,    @#20    ;RESET IOT VECTOR
017576  013700  001330                  MOV     DEVADR, R1      ;GET FIRST DJ ADR
017602  012711  340400                  MOV     #40400, (R1)+   ;SET CSR
                                                                ;BIT8 = TRANS SCAN ENABLE
                                                                ;BIT14 = TRANS INTERUPT ENABLE
017606  005721                          TST     (R1)+           ;INC POINTER
017610  012721  000001                  MOV     #1,     (R1)+   ;SET TCR, LINE 0
017614  000001                          WAIT                    ;WAIT FOR AN INTERUPT
017616  012637  000020                  MOV     (SP)+,  @#20    ;RESTORE IOT VECTOR
017622  000207                          RTS     PC

017624  162716  000010          4$:     SUB     #10,    (SP)    ;REPOSITION ADDRESS TO RVC VEC
017630  011637  001232                  MOV     (SP),   VECADR  ;SAVE FIRST VECTOR
017634  062706  000006                  ADD     #6,     SP      ;RESET STACK FROM IOT
017640  000002                          RTI                     ;RETURN FROM INTERUPT - RESTORE STATUS

017642  022737  000176  001316  KBDINT: CMP     @#SWREG,SWR
017650  001016                          BNE     1$
017652  005037  017706                  CLR     TMP1
017656  113737  177562  017706          MOVB    177562,TMP1
017664  042737  000200  017706          BIC     #200,TMP1
017672  122737  000007  017706          CMPB    #7,TMP1
017700  001002                          BNE     1$
017702  004737  017770                  JSR     PC,CNTLU
017706  000207          1$:             RTS     PC

017710  000000          TMP1:   0

017712  013746  000006          SUSWRR: MOV     6.-(SP)
017716  013746  000004                  MOV     4.-(SP)
017722  012737  017742  000004          MOV     #1$,4
017730  022777  177777  161362          CMP     #-1,@SWR
017736  001402                          BEQ     2$
017740  000407                          BR      3$
017742  022626                  1$:     CMP     (SP)+,(SP)+
017744  012737  000176  001316  2$:     MOV     #SWREG,SWR
017752  012737  000174  001320          MOV     #DISPREG,DISPLAY
017760  012637  000004          3$:     MOV     (SP)+,4
017764  012637  000006                  MOV     (SP)+,6
017770  000207                          RTS     PC

017772  022737  000176  001316  CNTLU:  CMP     #SWREG,SWR
020000  001023                          BNE     1$
020002  000004  020062                  TYPE    ,SWREG
020006  013705  000176                  MOV     SWREG,TTY       ;TYPE SWREG IN OCTAL
020012  004737  016562                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
020016  000004  020050                  TYPE    ,NEWIS
020022  004537  016216                  JSR     R5,READIN
020026  017706                          .WORD   TMP1
020030  001360                          BNE     CNTLU
020032  022737  000020  020050          CMP     #20,CNT
020040  001403                          BEQ     1$
020042  013777  017706  161350          MOV     TMP1,@SWR
```

```
 4040  020046  000207                  18:     R/S     PC
 4041
 4042
 4043  020050  000000                  CNT:    0
 4044
 4045  020052  020240  042516  036527  NEW18:  .ASCIZ  "  NEW= "
 4046  020070  020260  053523  036522  SWR8:   .ASCIZ  <15><12>"SWR= "
 4047  020076  000000
 4048
 4049          000001                  .END
```

```
AUTO    017470          596    4158#
BCSR    001214          542#   3457*  3458   3463*  3464   3484*  3539*  3563*  3635*  3655*  3671
BEGIN   001324          530     571#
BELL  = 000007          496#   3772   3893
BITYP$  016552         4005#
BIT0  = 000001          497#    967    969    979   1111   1133   1201   1219   1237   1255   1293   1321   1349
                       1377    1405   1433   1477   2812   2882   2946   2995   3045   3090   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   3379   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
CLRVEC  002336          727     729    732#
CNT     020050         3940*   3959*  4237   4242#
CNTLU   017770          730    4208   4229#  4236
CONFIG  001766          629     660#   675
CSR     001210          539#    738*   745*   750    779    780*   781*   800*   801    805    809*   810    821*
                        822     826    830*   831    842*   843    847    851*   852    863*   864    868    872*
                        873     884*   885    889    893*   894    905*   906    910    914*   915    926*   927
                        931     935*   936    948*   950*   951    967*   968    969    973    979*   980    981
                        994*    995*   996   1001*  1002*  1003*  1004*  1005   1019*  1020   1024*  1025  1029*
                       1030    1110*  1113   1129   1134   1149*  1154   1174   1179   1200*  1203   1210*  1212
                       1220    1236*  1238   1249*  1264*  1266   1277*  1292*  1294   1305*  1320*  1322   1333*
                       1348*   1350   1361*  1376*  1378   1389*  1404*  1406   1417*  1432*  1434   1445*  1478
                       1484    1488   1527   1533*  1561   1567   1571   1610   1616*  1644   1650   1654   1693
                       1699*   1727   1733   1737   1776   1782*  1812   1818   1822   1861   1867*  1895   1901
                       1905    1944   1950*  1978   1984   1988   2027   2033*  2061   2067   2071   2110  2116*
                       2146    2152   2156   2195   2201*  2229   2235   2239   2278   2284*  2312   2318
                       2361    2367*  2395   2401   2405   2444   2450*  2480   2486   2490   2529   2535*
                       2569    2573   2612   2618*  2646   2652   2656   2695   2701*  2729   2735   2739
                       2784*   2814   2820   2823*  2824   2830   2843   2860*  2883   2886   2889   2896
                       2931*   2941*  2944   2947   2953   2966*  2967   2996   3002   3015*  3046   3049
                       3091    3094   3113*  3137   3148   3152   3180   3227   3244*  3245   3251   3254
                       3287    3302   3308   3311   3316*  3319   3340   3365*  3376*  3379   3383   3386
                       3396*   3397   3403   3430*  3436   3446*  3456*  3488   3566   3640*  3654*  3661  3697*
                       3698*   3703   3708   3715   3758*  3817*  3820*  3823*  3826*  3838   3841*  3919

DEFADR  001312          560#    605
DEFVEC  001314          561#
DEVADR  001230          550#    601    603    605*   606*   607    745   4172*  4195
DISPLA  001320          564#   3869*  3873*  4222*
DISPRE  000174          526#   4222
DJLEN   001306          558#    740*   747*  1511   1594   1677   1760   1784*  1845   1928   2011   2094  2118*
                       2179    2262   2345   2428   2462*  2513   2596   2679   2762   2786*  3730*  3748  3754*

DJMXNO= 000020          515#    655
DJPAR   001310          559#    741*   748*  3510   3596
DJUUT   001304          557#    658*   742    744*   749*   3762
```

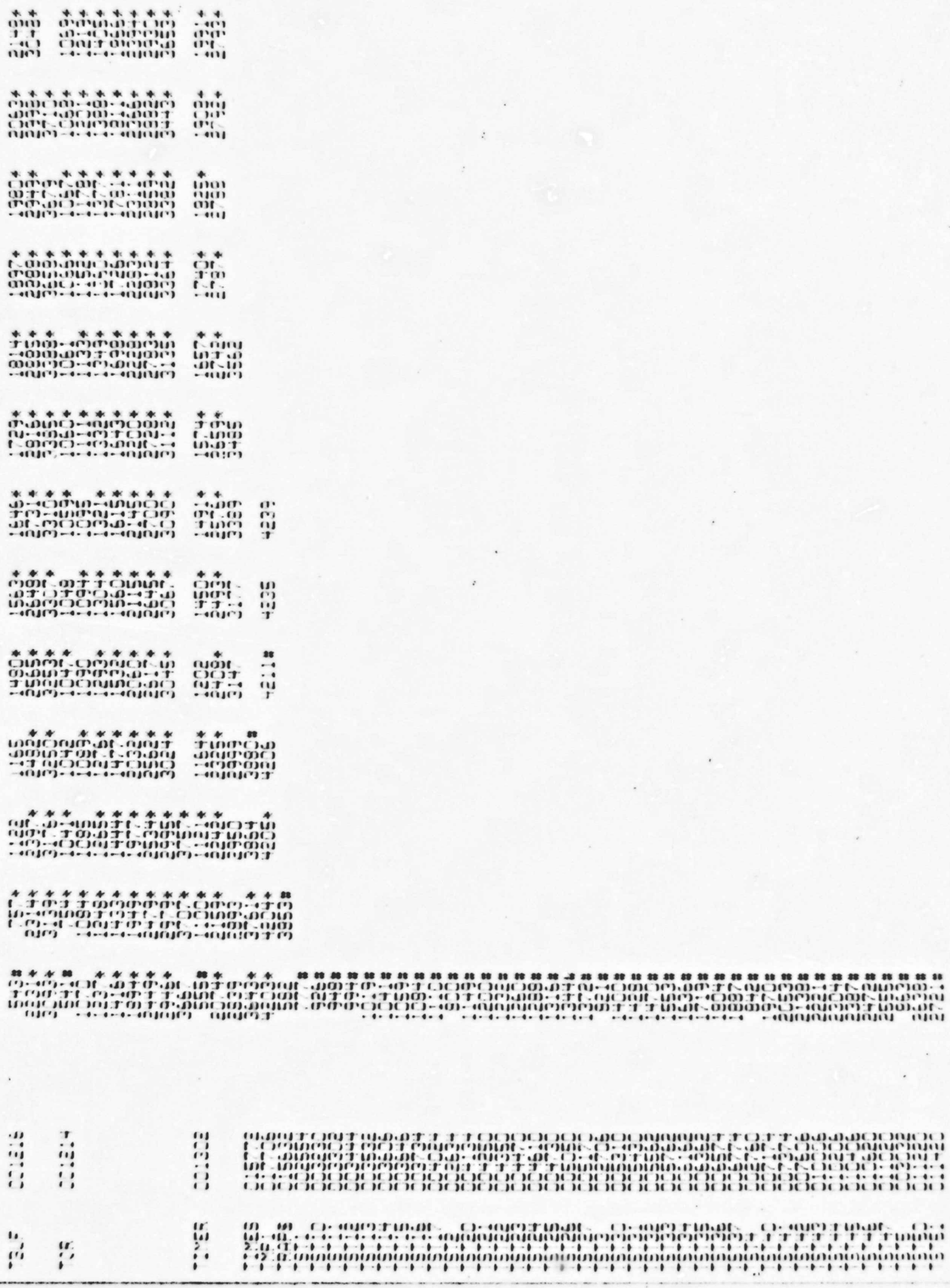
```
DONE    015410          3763    3766#   4179
EMT$    015774           580    3890#
END22   004434          1240    1243    1246#
END23   004554          1268    1271    1274#
END24   004674          1296    1299    1302#
END25   005014          1325    1327    1330#
END26   005134          1353    1355    1258#
END27   005254          1381    1383    1386#
END30   005374          1409    1411    1414#
END31   005514          1437    1439    1442#
END56   012630          3051    3054    3057#
END57   013000          3097    3099    3102#
END66   015330          3733    3754#
ERRORS  001200           536#    591*   3894*   3915
ERROR$  016146          3904    3918#
FTIME   001320           567#    726     731*
GETADR  001506           595     599#    602     609
GETLEN  002104           689#    694     698     702
GETNUM  001632           622#    631     633     638     640     643     656
GETPAR  002000           692     704#    713
GETVEC  001564           610#    613     619     621
HLT     = 104000         484#    783     788     803     807     812     824     828     933     845     849     954     866
                         870     875     887     891     896     908     912     917     929     933     938     954     960
                         964     971     975     983     999    1008    1022    1027    1032    1045    1050    1064    1073
                        1087    1092    1097    1117    1127    1131    1136    1158    1169    1176    1181    1205    1216
                        1222    1242    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    2328
                        2332    2342    2348    2354    2359    2364    2406    2411    2415    2425    2431    2437    2442
                        2447    2491    2496    2500    2510    2516    2522    2527    2532    2574    2579    2583    2593
                        2599    2605    2610    2615    2657    2662    2666    2676    2682    2688    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    3571
                        3577    3583    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#
HLTCT$  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    2638    2721    2804    2875    3128    3826#
INITR   015574          3818    3821    3824    3828#
INPUT   016512           625     662     670     672     674     691     693     695*    699*    706     709     710     712
```

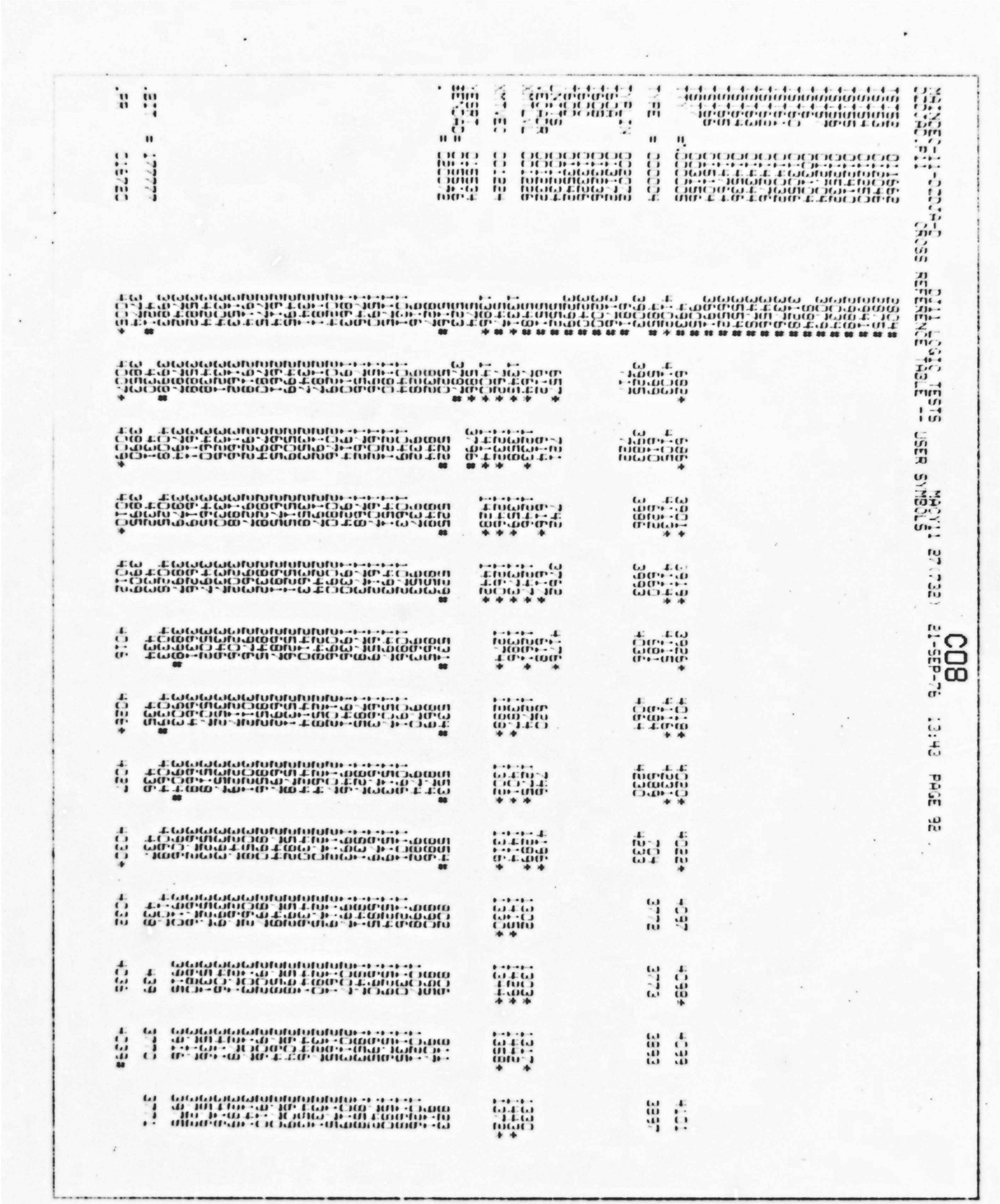
```
                    3941    3971    3972    3998#
IOTRAP  017104       576    4077#
IOT$    017130      4078    4097#
ISR22   004424      1232    1242#
ISR23   004544      1260    1270#
ISR24   004664      1288    1298#
ISR25   005006      1316    1327#
ISR26   005126      1344    1355#
ISR27   005246      1372    1383#
ISR30   005366      1400    1411#
ISR31   005506      1428    1439#
ISR56   012620      3034    3053#
ISR57   012774      3079    3099#
ISR66   015274      3705    3735#
KBDINT  017640      3767    3853    3890    3908    4201#
KIT$    015742      3859    3865    3872#   3913
LAD     015770       766*   1101*   1451*   1785*   2119*   2453*   2794*   3868*   3874    3876    3879#
LENGTH  001236       553#    665     678     747
LEVEL7= 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#    988#   1012#   1035#
                    1053#   1077#   1100#   1102    1141#   1192#   1226#   1254#   1282#   1310#   1338#   1366#   1394#
                    1422#   1450#   1452    1535#   1618#   1701#   1784#   1786    1869#   1952#   2035#   2119#   2130
                    2203#   2286#   2369#   2452#   2454    2537#   2620#   2703#   2786#   2795    2864#   2934#   2982#
                    3028#   3073#   3118#   3370#   3450#   3469#   3545#   3643#   3644    3685#   3761#
NEWIS   020052      4233    4244#
OPEN    = 000000     514#
OVER$   015746      3857    3873#
PARITY  001276       554#    679     718    3510    3596
PC      =%000007     495#    572*    596*    623*    661*    683*    687*    690*    705*    730*   1470*   1553*   1636*
                    1719*   1804*   1887*   1970*   2053*   2138*   2221*   2304*   2387*   2472*   2555*   2638*   2721*
```

```
                    2804*  2875*  2888*  3038*  3083*  3128*  3478*  3554*  3662   3767*  3842*  3853*  3890*
                    3902*  3904*  3908*  3920*  3929*  3934*  3995*  4038*  4193*  4208*  4209*  4225*  4232*
                    4240*
PCNT    001204       537#   592*   593*  3768*  3769*  3794
PDOWN$  016744       578   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*  1431*  1446*  3036*
                    3037*  3069*  3081*  3082*  3114*  3240*  3241*  3299*  3300*  3366*  3652*  3694*  3695*
                    3961   3965*
PSTEMP  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   3266   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
READS$  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*  1908   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   1518   1523*  1527*  1528   1561*  1571*  1575*
                    1579   1586   1595   1601   1606*  1610*  1611   1644*  1654*  1658*  1662   1669   1678
                    1684   1689*  1693*  1694   1727*  1737*  1741*  1745   1752   1761   1767   1772*  1776*
                    1777   1812*  1822*  1826*  1830   1837   1846   1852   1857*  1861*  1862   1895*  1905*
                    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   2146*  2156*  2160*  2164   2171   2180   2186   2191*  2195*  2196   2229*
                    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*  2844   2851   2855*  2893*
                    2886*  2889*  2896*  2901   2905*  2909   2913*  2917   2921*  2925*  2926   2947*  2953*
```

DZDJAD.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

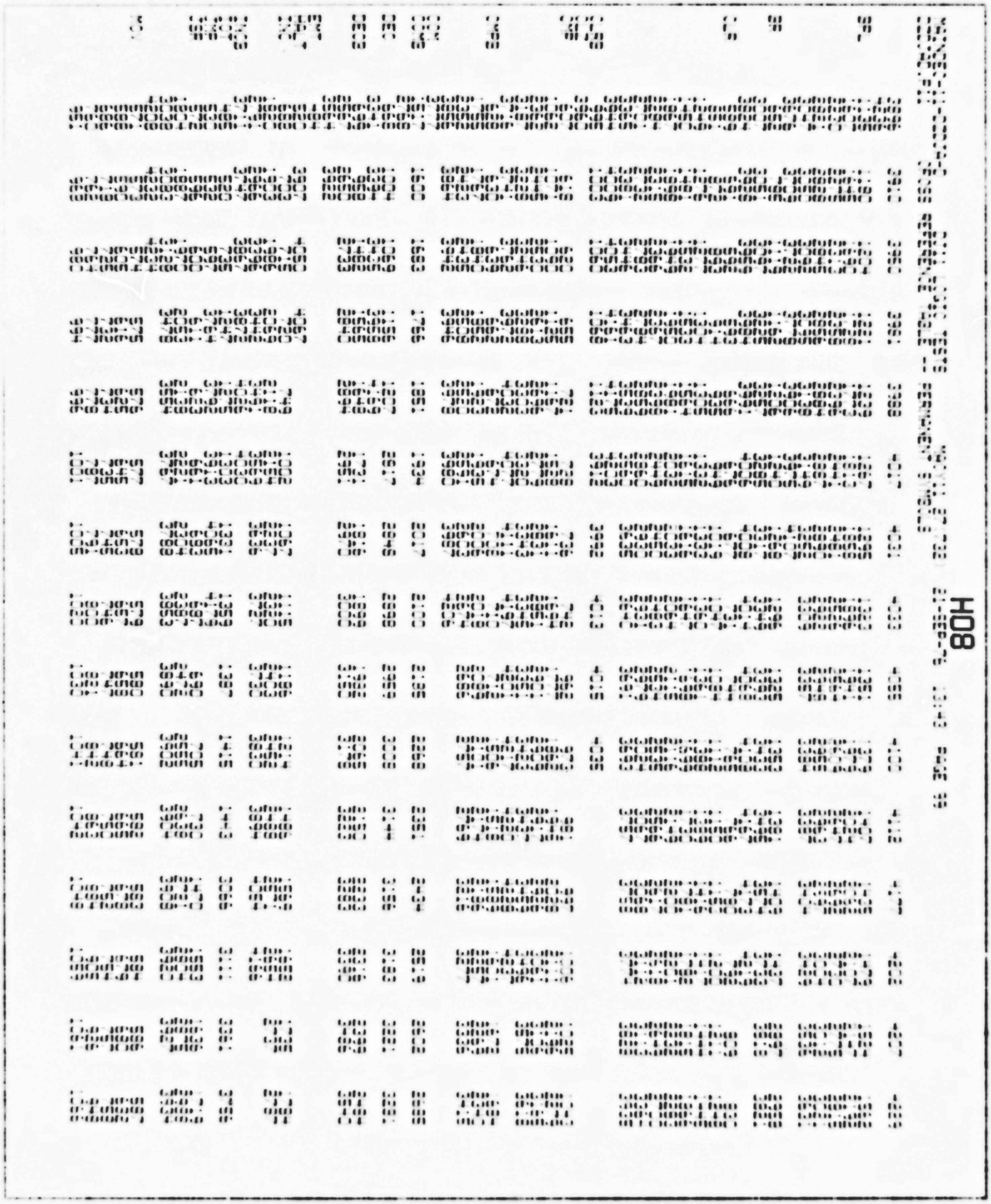
```
                    2954    2962*   2967*   2968    2975*   2996*   3002*   3003    3011*   3020*   3046*   3057*   3061
                    3091*   3102*   3106    3137*   3140    3148*   3152*   3161    3165    3180*   3181    3187    3201*
                    3205    3213    3216    3227*   3228    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*   3953*   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    1837*   1838*
                    1839*   1840    1845*   1846    1851*   1852    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*   3587    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*   1899*   1982*   2065*
                    2150*   2233*   2316*   2399*   2484*   2567*   2650*   2733*   2840*   2894*   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    2202    2285    2368    2451
                    2536    2619    2702    2785    2862    2932    2981    3026    3071    3116    3368    3448    3468
                    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*   3833    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    3891    3895    3905    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
```

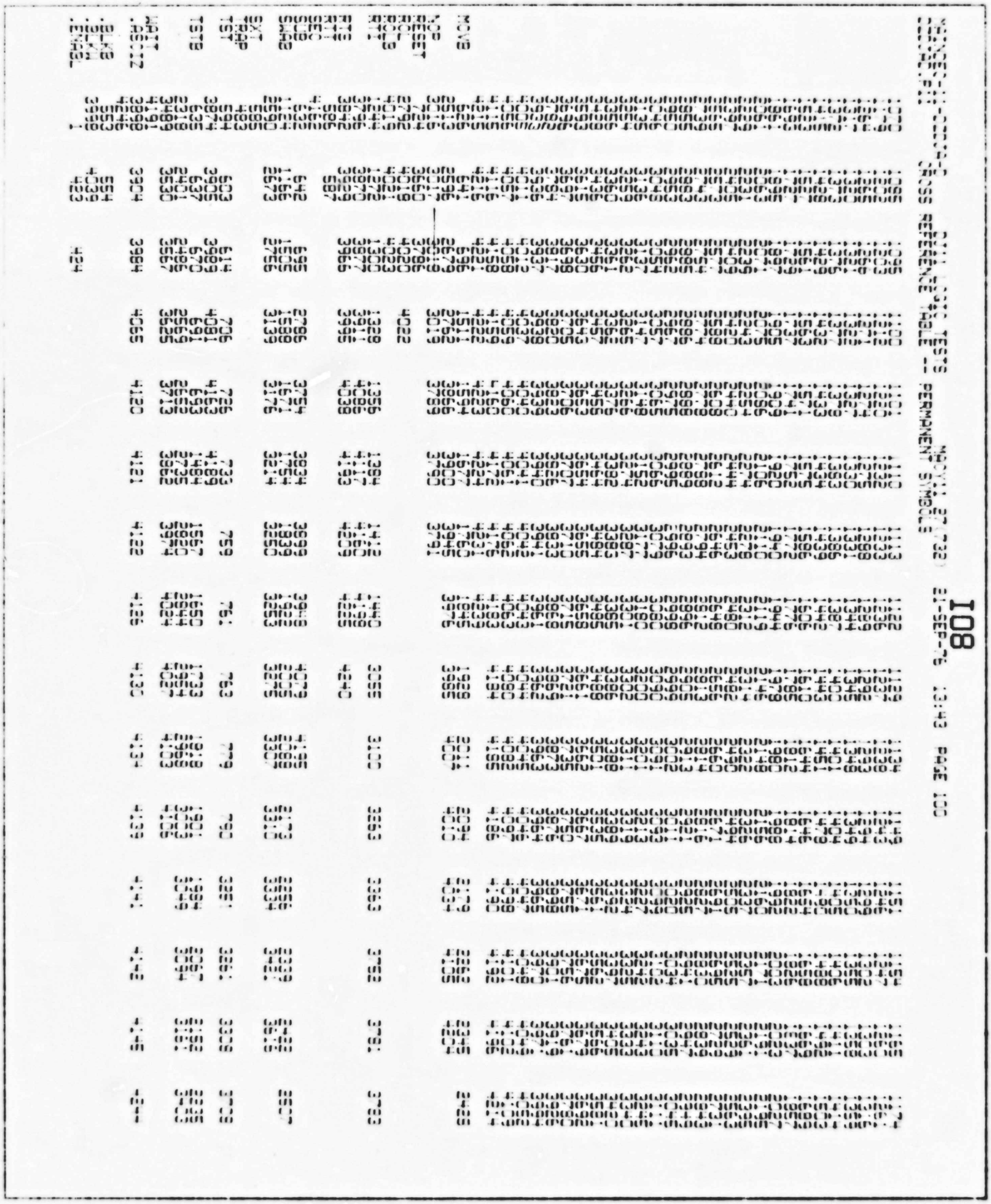


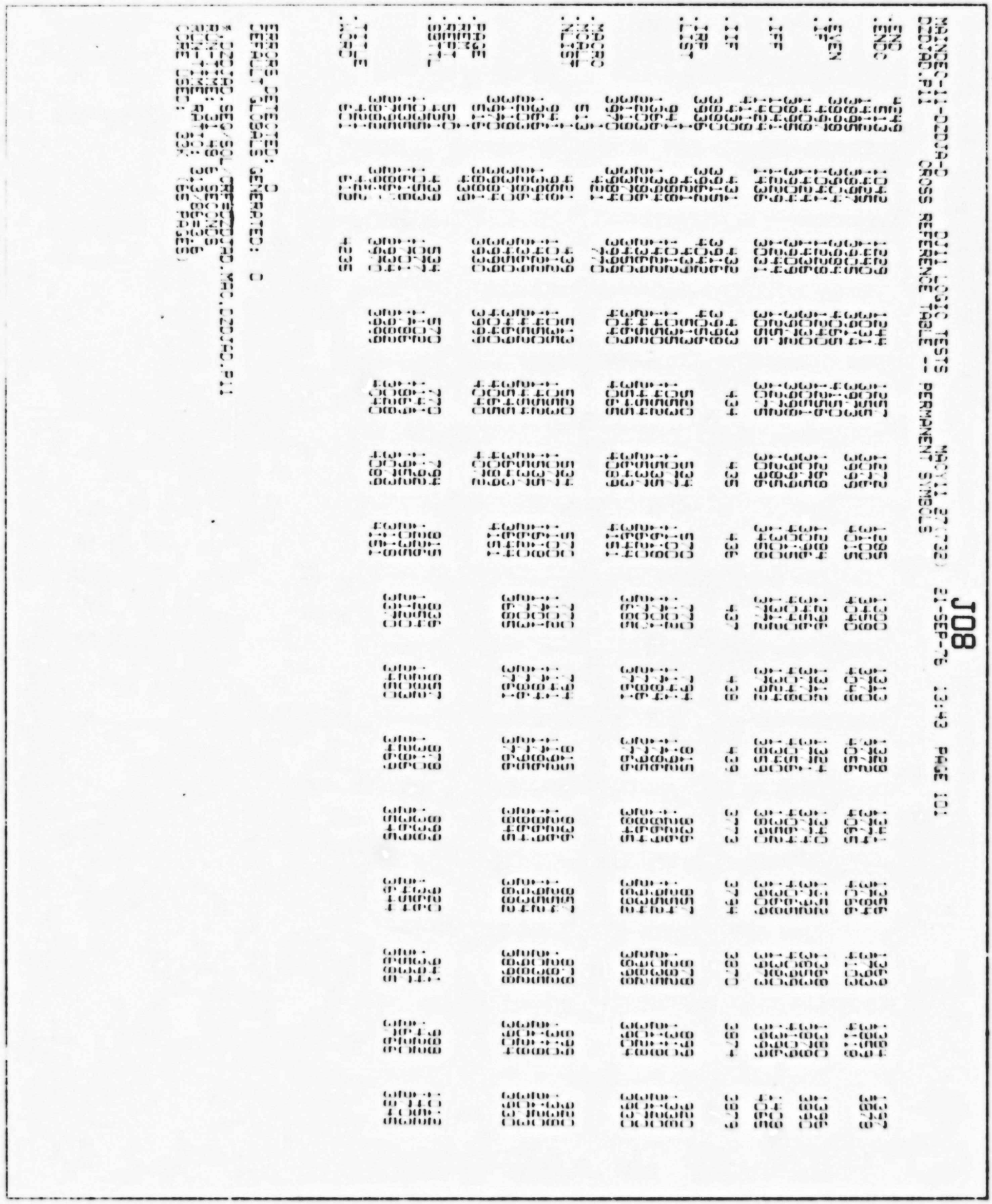


```
.BRF    016640          4014    4017    4032#
.BRL    016616          4015#   4031
.PIT    016604          4006    4011#
.SAVR0  016600          4048#   4053    4070#
.SCNT   016636          3928#   3994#
.TYPE   017234          3965#   3966    4101#   4103    4118#
```

```
BITYPE        1#    513#
BLKBLK        1#
BUFHDR        1#
DUMP          1#    513#    3901    3919    4231
DUMP18        1#
FILBLK        1#
LNKBLK        1#
ODT11         1#
ODT11X        1#
POP           1#    513#    3962    4056
PRINT         1#    513#    3897    3903    3982    4064
PUSH          1#    513#    3936    4042
SCOPE.        1#
SCOP.         1#
SDUMP         1#    513#     682     686
TRACE         1#
TRNBLK        1#
TYPEM         1#
WRITE         1#
$BRK        770#    3545
$BRKO       770#    3469
$BYTRF      770#     989    1053
$CATCH        1#
$CLREG      770#    1012    1077
$CMOS       770#    2795
$CMTAG        1#
$CNTL         1#    4227
$CSREF      770#
$DONE       770#    1452    1535    1618    1701    1786    1869    1952    2035    2120    2203    2286    2369    2454    2537
            2620    2703
$END          1#    [illegible]
$EQUAT        1#    [illegible]
$EXERR      770#    [illegible]
$EXTOV      770#    [illegible]
$HD         770#    [illegible]
$HLT          1#    [illegible]
$IOCAT      461#    [illegible]
$IOT        461#    4072
$KBIN         1#    4200
$KRAT         1#
$LOADR        1#
$LV         770#    1226    1254    1282    1310    1338    1366    1394    1422
$MTSE       770#    1198
$OCTAL        1#    3998
$POWER        1#    4040
$RAND         1#
$RANCH        1#
$RCVDT      770#    2934
$RCVE       770#     941
$READ         1#    3970
$REGISC     770#     794     815     836     857     878     899     920
$RESET      770#    1036    3460
$RESULT     770#    3644
$RINT       770#    3028    3073
$RONLY      770#
$RUOR       770#
```







K08

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