Micro Fiche Scan

Name of device(s) tested:
    DJ11
Test description:
    DJ11 MODULE
MAINDEC Number or Package Identifier (after SEP 1977):
    CXDJAL0
Fiche Document Part Number:
    AH-E723L-MC
Fiche preparation date unknown, using copyright year:
    1982
Image resolution:
    8-bit gray levels, max. quality for archiving

```
.REM %
```

IDENTIFICATION

PRODUCT CODE: AC-E721L-MC

PRODUCT NAME: CXDJAL0 DJ11 MODULE

DATE: JUNE 1981

MAINTAINER: DEC/X11 SUPPORT GROUP


```
%
```

```
.REM %
1.      ABSTRACT
        --------
        THIS MODULE EXERCISES ANY COMBINATION OF ONE TO SIXTEEN LINES ON ANY
        COMBINATION OF FROM ONE TO FOUR DJ11 ASYNCHRONOUS, SIXTEEN LINE
        COMMUNICATION DEVICES.  65 CHARACTERS ARE TRANSMITTED ON EACH
        DJ11 UNDER INTERRUPT CONTROL. TRANSMISSION OF THE 65TH CHARACTER
        CAUSES A RECEIVER BUFFER FULL INTERRUPT ON THE DJ11. THE ROUTINE
        DISABLES INTERRUPTS ON THE RESPECTIVE DJ11 AND EMPTIES THE HARDWARE
        SILO INTO A SOFTWARE BUFFER.  THE DATA IS CHECKED, THEN INTERRUPTS ARE
        REENABLED AND TRANSMISSION RESUMED FOR THE GIVEN DJ11.  AS EACH
        LINE FINISHES TRANSMISSION OF THE FULL 8-BIT COUNT PATTERN, IT IS
        DISABLED. THUS, THERE ARE TIMEW WHEN ONLY A SUBSET OFTHE SELECTED LINES
        ARE ACTUALLY RUNNING. WHEN ALL LINES ON A DJ11 HAVE TRANSMITTED THE
        ENTIRE COUNT PATTERN, THEY ARE ALL REENABLED AND THE PATTERN TRANSMISSION
        IS RESTARTED. A PASS IS DEFINED AS 200 (OCTAL) TRANSMISSIONS OF 65
        CHARACTERS EACH ON EACH ACTIVE DJ11. THE MODULE ASSUMES THAT ALL SIXTEEN
        LINES WILL BE EXERCISED AND WILL BE SENDING 8 BIT CHARACTERS.  SEE
        "OPERATION OPTIONS" BELOW FOR NECESSARY CHANGES REQUIRED FOR OTHER
        CONFIGURATIONS.
        A BACKGROUND "WATCHDOG TIMER" IS ALSO RUN BY THE MODULE TO ALLOW
        DETECTION OF THE "DEVICE HUNG" CONDITION (AN EXPECTED INTERRUPT THAT FAILED
        TO OCCUR). THE TIMER IS SET TO RUN LONGER THAN THE LONGEST POSSIBLE
        MODULE PASS TIME, AND ON A HEAVILY LOADED SYSTEM WILL IN FACT
        TAKE A LONG TIME TO TIMEOUT (I.E. 10-15 MINUTES). WHEN THE "DEVICE
        HUNG" CONDITION IS DETECTED, A "DEVICE HUNG" MESSAGE IS OUTPUT AND
        THE MODULE IS THEN DROPPED. THIS PREVENTS A LONG RUN FROM BEING
        PERMANENTLY UNABLE TO RELOCATE DUE TO A "LOST" INTERRUPT.

2.      REQUIREMENTS
        ------------
        HARDWARE: DJ11 SIXTEEN LINE ASYNCHRONOUS COMMUNICATION DEVICE.
        STORAGE:: DJA REQUIRES:
                        1. DECIMAL WORDS: 01169
                        2. OCTAL WORDS:   02217
                        3. OCTAL BYTES:   04436


3.      PASS DEFINITION
        ---- ----------
        ONE PASS OF THE DJA MODULE CONSISTS OF TRANSMITTING,RECEIVING,AND
        CHECKING 8320. CHARACTERS PER DJ11.

4.      EXECUTION TIMES
        --------- -----
        WORST CASE (1 LINE ONLY ON EACH OF 4 DJ11'S AT 75 BAUD) TAKES
        APPROXIMATELY 3 MINUTES PER PASS. RUNNING ALL LINES (DEFAULT CASE)
        WILL CAUSE SEVERAL PASSES PER MINUTE TO BE COMPLETED.

5.      CONFIGURATION REQUIREMENTS
        ------------- ------------
        DEFAULT PARAMETERS:
                BR1: 5
                DEVICE COUNT(DVID1): 1

        REQUIRED PARAMETERS:
```

FIRST DEVICE ADDRESS: THE FIRST DEVICE REGISTER ADDRESS
MUST BE SPECIFIED.
VECTOR: THE VECTOR ADDRESS OF THE FIRST DJ11 MUST BE GIVEN
DVID1: IF MORE THAN 1 UNIT IS BEING RUN, VALUE MUST BE GIVEN
(4 UNITS MAX)

6. DEVICE/OPTION SETUP

ALL SIXTEEN LINES OPERATIONAL (DEFAULT) OR ALL SELECTED LINES OPERATIONAL
(IF NOT ALL LINES SELECTED).

7. MODULE OPERATION

A. INITIALIZE FIRST VALUES TO BE TRANSMITTED FRO EACH LINE
(SET TO ZERO).

B. INITIALIZE INTERNAL QUEUES AND PARAMETERS, SETUP LINKAGE OF DEVICE
VECTORS TO LINKAGE TABLE, AND INITIALIZE SELECTED DJ11'S
VIA SETTING "MOS CLEAR".

C. SET TRANSMITTER INTERRUPT ENABLE AND SILO FULL INTERRUPT ENABLE,
THEN SET TRANSMITTER ENABLE FOR ALL SELECTED LINES ON EACH
SELECTED DJ11. COUNT DJ11'S SELECTED AS THEY GET TURNED ON.

D. INITIALIZE WATCHDOG TIMER, WHICH RUNS VIA A SET OF "BREAK"
LOOPS. THE LOOP MONITORS THE COUNT OF ACTIVE DJ11'S. IF THE DJ11
COUNT GOES TO ZERO BEFORE THE TIMER TIMES OUT, AN END OF PASS CALL
IS MADE. (WHICH WILL EVENTUALLY CAUSE THE MODULE TO BE RESTARTED
AT STEP B). IF THE TIMER TIMES OUT BEFORE THE DJ11 ACTIVE
COUNT GOES TO ZERO, A "DEVICE HUNG" MESSAGE IS OUTPUT AND
THE MODULE IS DROPPED.

E. WHEN A TRANSMIT INTERRUPT OCCURS, TRANSMIT INTERRUPT ENABLE IS
TURNED OFF FOR THAT DJ11, A UNIT IDENTIFICATION IS SAVED IN
THE MODULE'S TRANSMIT QUEUE, AND THE MODULE PIRQ'S. WHEN THE MONITOR
RETURNS TO THE TRANSMIT ISR, THE NEXT CHARACTER FOR THE
INTERRUPTING LINE IS TRANSMITTED. IF THE CHARACTER TRANSMITTED
WAS 377 (OCTAL), THEN THAT LINE IS DISABLED FROM BEING
RUN UNTIL ALL THE OTHER LINES HAVE ALSO REACHED THE MAX VALUE
(ACTUALLY, IT WON'T BE DISABLED UNTIL THE NEXT 65 CHARACTER
TRANSMISSION ON THIS DJ11). IF ALL LINES ARE DISABLED,
THE SELECTED LINES GET REENABLED. IF 65 (DECIMAL) CHARACTERS
HAVE BEEN TRANSMITTED, THE TRANSMITTER IS DISABLED BY CLEARING THE
TRANSMITTER CONTROL REGISTER. IF LESS THAN 65 CHARACTERS HAVE
BEEN TRANSMITTED, TRANSMIT INTERRUPT ENABLE IS REENABLED. THE
TRANSMITTER SERVICE THEN EXITS.

F. WHEN 65 CHARACTERS HAVE BEEN TRANSMITTED ON A GIVEN DJ11, A
RECEIVER BUFFER ("SILO") FULL INTERRUPT SHOULD OCCUR. THE
MODULE THEN DISABLES THE "SILO FULL" INTERRUPT, QUEUES A
UNIT IDENTIFICATION IN IT'S RECEIVER QUEUE, AND PIRQ'S. WHEN THE
MONITOR RETURNS TO THE RECEIVER SERVICE, THE MODULE DUMPS
THE CONTENTS OF THE HARDWARE SILO OF THE INTERRUPTING DJ11 INTO A
SOFTWARE BUFFER AND CHECKS THE DATA. IF THE REQUIRED NUMBER OF
TRANSMISSIONS (STORED IN ENDSEL) HAVE BEEN PERFORMED BY THIS

DJ11, IT IS TURNED OFF AND THE DJ11 ACTIVE COUNT IS
DECREMENTED. IF NOT ALL DONE, THE RECEIVER DATA TABLE IS RESYNCED
TO MATCH THE TRANSMITTER TABLE AND THE DJ11 IS REENABLED TO
TRANSMIT ANOTHER 65 CHARACTERS. WHEN THE DJ11 ACTIVE COUNT
GOES TO 0, THE WATCHDOG TIMER WILL DETECT IT AND ISSUE AN
END OF PASS (SEE STEP D).

THE SELECTED LINES GET REENABLED. IF 65 (DECIMAL) CHARACTERS

8. OPERATION OPTIONS

A. TO MAKE THE TIME INTERVAL BETWEEN ENDPAS STATEMENTS LONGER OR SHORTER,
EITHER INCREASE OR DECREASE THE 'ENDSEL' WORD.

B. TO EXERCISE A CONFIGURATION OF LINES OTHER THAN ALL SIXTEEN AT ONCE,
CHANGE THE APPROPRIATE BITS OF THE 'XMITEN' WORD FOR THE SELECTED DJ11
AND THE SAME BITS OF THE 'XMTSEL' WORD. A ONE INDICATES THAT
THE LINE IS TO BE EXERCISED, A ZERO INDICATES THAT THE LINE IS OFF.

C. MEANING OF SR1

1. FOR 5 BIT DATA SET BIT 2 (SR1=000004)
   FOR 6 BIT DATA SET BIT 1 (SR1=000002)
   FOR 7 BIT DATA SET BIT 0 (SR1=000001)
   FOR 8 BIT DATA SET NO BITS (SR1=000000)
   DEFAULT VALUE IS 8 BIT DATA LENGTH
   CHECH HARDWARE STRAPPING TO DETERMINE SR1 SETTING
2. ITERATION COUNT - USED TO ADJUST PASS TIME TO BAUD RATE
   FOR 75 BAUD SET BIT 5
   FOR 110 BAUD SET BIT 6
   FOR 134.5 BAUD SET BIT 7
   FOR 150 BAUD SET BIT 8
   FOR 300 BAUD SET BIT 9
   FOR 600 BAUD SET BIT 10
   FOR 1200 BAUD SET BIT 11
   FOR 1800 BAUD SET BIT 12
   FOR 2400 BAUD SET BIT 13
   FOR 4800 BAUD SET BIT 14
   FOR 9600 BAUD SET BIT 15 OR NO BITS (5-15=0)

9. NON STANDARD PRINTOUTS

THERE ARE 2 SPECIAL MESSAGES WHICH MAY BE PRINTED OUT. THEY ARE:

'UNIT DROPPED''- THIS INDICATES THAT A DJ11 WHICH HAS JUST PREVIOUSLY
REPORTED A FAILURE HAS BEEN DROPPED. DJ11'S ARE DROPPED IF
''BUSY CLEAR'' FAILS TO CLEAR WHEN THEY ARE INITIALIZED.

'UNIT HUNG''- THIS INDICATES THAT THE MODULE'S WATCHDOG TIMER
HAS TIMED OUT BEFORE THE DJ11 DEVICE ACTIVE COUNT WENT TO ZERO.
THIS OCCURS WHEN AN EXPECTED TRANSMITTER OR RECEIVER INTERRUPT
FAILS TO OCCUR ON ONE OR MORE DJ11'S.

```
*****************************************************************************
*                                                                           *
*       EDIT:   BY:             DATE:           REASON:                     *
```

```
;*                                                                              *
;*      003     R. GAUDIN       19-JUN-81       ADD ERROR PRINTOUT WHEN THE     *
;*                                              VECTOR + ADDR LOCS. ARE ZERO.   *
;*                                                                              *
;********************************************************************************


.SBTTL  %
```

```
                        .SBTTL MODULE HEADER
000000'                 IOMOD  <DJAL >,0,0,5,,,0,36
000000'                 MODULE  140000,DJAL ,0,0,5,,,0,36,,,,,
                        .TITLE  DJAL DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6.2    23-APR-81
                                .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 045104  046101     040  MODNAM: .ASCII  /DJAL / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 000000                  ADDR:   0+0             ;1ST DEVICE ADDR.
000010' 000000                  VECTOR: 0+0             ;1ST DEVICE VECTOR.
000012'    240                  BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'    000                  BR2:    .BYTE   PRTY+0          ;2ND BR LEVEL.
000014' 000001                  DVID1:  +1              ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN            ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN            ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN            ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN            ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 140000                  STAT:   140000          ;STATUS WORD.
000030' 002056'                 INIT:   START           ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP           ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0               ;PASS COUNTER.
000036' 000000                  ICONT:  0               ;# OF ITERATIONS PER PASS=0
000040' 000000                  ICOUNT: 0               ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0               ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0               ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0               ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0               ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0               ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0               ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                 ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0               ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0               ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN            ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN            ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN            ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN            ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN            ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN            ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN            ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN            ;ADDR OF CURRENT CSR.
000102'                         SBADR:                  ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN            ;CONTENTS OF CSR.
000104'                         WASADR:                 ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN            ;STATUS REG CONTENTS.
000106'                         ERRTYP:                 ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN            ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN            ;ACTUAL DATA.
000112' 002440'                 RSTRT:  RESTRT          ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN            ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN            ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN            ;# OF INTERRUPTS PER ITERATION
000122' 000036                  IDNUM:  36              ;MODULE IDENTIFICATION NUMBER=36
        000040                          .REPT   SPSIZ   ;MODULE STACK STARTS HERE.
```

```
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
      000224'                           MODSP:
                                        ;*****************************************************************************

                                        ;+
                                        ; THE NEXT LOCATION  CONTAINS A NUMBER  THAT CAN BE EXAMINED IN CORE TO  BE
                                        ; SURE THAT THE LISTING AND ACTUAL CODE ARE THE SAME.  THIS  LOCATION IS TO
                                        ; BE INCREMENTED  EACH TIME THE SOURCE CODE IS UPDATED.  LOCATION  'MODREV'
                                        ; IS NOT USED BY THE PROGRAM.
                                        ;-

      000224' 000005                    MODREV: 1+1+1+1+1+0

                                        ;*****************************************************************************

                                        .SBTTL SOME VARIABLES AND CONSTANTS UNIQUE TO THIS ROUTINE
      000226' 004567  003346            DJLINK: JSR     R5,RCVINT               ;UNIT 0 RECEIVER
      000232' 000000                            0
      000234' 004567  003004                    JSR     R5,XMTINT               ;UNIT 0 TRANSMITTER
      000240' 000000                            0
      000242' 004567  003332                    JSR     R5,RCVINT               ;UNIT 1 RECEIVER
      000246' 000010                            10
      000250' 004567  002770                    JSR     R5,XMTINT               ;UNIT 1 TRANSMITTER
      000254' 000010                            10
      000256' 004567  003316                    JSR     R5,RCVINT               ;UNIT 2 RECEIVER
      000262' 000020                            20
      000264' 004567  002754                    JSR     R5,XMTINT               ;UNIT 2 TRANSMITTER
      000270' 000020                            20
      000272' 004567  003302                    JSR     R5,RCVINT               ;UNIT 3 RECEIVER
      000276' 000030                            30
      000300' 004567  002740                    JSR     R5,XMTINT               ;UNIT 3 TRANSMITTER
      000304' 000030                            30

              000000                            OPEN=0

      000306' 000040                    XMTTBL:.BLKW 32.                ;A TABLE FOR RECORDING CHARACTERS WHICH HAVE
                                                                        ;BEEN TRANSMITTED
      000406' 000100                    RCVTBL:.BLKW 64.                ;A TABLE FOR RECORDING CHARACTERS WHICH YOU
                                                                        ;EXPECT TO RECEIVE
      000606' 000404                    SILO:   .BLKW 260.              ;A BUFFER AREA TO STORE THE DATA RECORDED IN
                                                                        ;THE DJ'S RECEIVER SILO
      001616' 000004                    XQ:     .BLKW 4.                ;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
                                                                        ;TRANSMIT INTERRUPT IS IN 'PIRQ"
      001626' 000004                    RQ:     .BLKW 4.                ;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
                                                                        ;RECEIVER INTERRUPT IS IN 'PIRQ"
      001636' 000020                    ERRQ:   .BLKW   16.
      001676' 000000                    ERRQI:  OPEN
      001700' 000000                    ERRQO:  OPEN

      001702' 000101  000101  000101    XCNT:.WORD 101,101,101,101      ;NUMBER OF CHARACTERS LEFT TO BE TRANSMITTED DURING
      001710' 000101                                                    ;THIS INTERRUPT. A WORD FOR EACH DEVICE
```

```
                                ;NUMBER OF CHARACTERS TO BE PROCESSED BY THE
                                ;RECEIVE ROUTINE. A WORD FOR EACH DEVICE

                                ;NUMBER OF ''65 CHARACTER CHUNKS''
                                ;TO BE PROCESSED BY THE MODULE PER PASS

                                ;LINES TO BE RUN

                                ;WORD USED TO FILL TRANSMITTER COMMAND
                                ;REGISTER 16XXX2,CORRESPONDS TO
                                ;A ONE FOR EACH ACTIVE LINE

                                ;SAVE LINES ACTIVATED IN CASE OF HANG

                                ;POINTERS USED FOR THE BOOKKEEPING OF
                                ;THE TRANSMIT QUEUE
                                ;POINTERS USED FOR THE BOOKKEEPING OF
                                ;THE RECEIVE QUEUE

                                ;TEMPORARY STORAGE LOCATION.USED DURING RECEIVE DATA CHE
                                ;MASK TO CHECK FOR UNIT SELECTED

                                ;USED DURING INITIALIZATION,THE ONE REPRESENTS THE
                                ;DONE BIT IN THE EXPECTED RECEIVE DATA TABLE
                                ;REFILL OF 'ENDSEL' WORD AFTER ENDPASS
                                ;REFILL OF 'RCNT' AND 'XCNT' AFTER AN INTERRUPT SERVICE
                                ;NOT VARIABLE SINCE 65. CHARS REQD TO GET SILO FULL INT

                                ;HOLDS VECTOR ADDRESS DURING SETUP ERROR PRINTOUT
                                ;HOLDS LINK ADDRESS DURING SETUP ERROR PRINTOUT

                                ;NUMBER OF DJ11'S STILL RUNNING
                                ;WATCHDOG TIMER OUTER LOOP VALUE


                                        ;-DATA BIT LENGTH (5-8) MASK
                                        ;-TEMP HOLDING FOR MASK & ITERATION COUNT &
                                        ;-DATA COMPARSION DURRING TEST.
                                        ;-9600  BPS VARIABLE
                                        ;-4800  BPS ITERATION
                                        ;-2400  BPS COUNTS
                                        ;-1800  BPS PER
                                        ;-1200  BPS LINE
                                        ;-600   BPS SPEEDS
                                        ;-300   BPS
```

```
001712' 000101  000101  000101  RCNT:.WORD 101,101,101,101
001720' 000101

001722' 000200  000200  000200  ENDCNT:.WORD 200,200,200,200
001730' 000200

001732' 177777  177777  177777  XMTSEL: .WORD   177777,177777,177777,177777
001740' 177777

001742' 177777  177777  177777  XMITEN:.WORD 177777,177777,177777,177777
001750' 177777

001752' 000000  000000  000000  XMITSV: .WORD 0,0,0,0
001760' 000000

001762' 001616'                 XPNTW:  XQ
001764' 001616'                 XPNTR:  XQ
001766' 001626'                 RPNTW:  RQ
001770' 001626'                 RPNTR:  RQ

001772' 000000                  I:      000
001774' 000001                  SELMSK: 1
001776' 000000                  SELECT: 0
002000' 100000                  EXPLIN: 100000

002002' 000200                  ENDSEL: 200
002004' 000101                  CNTSEL: 101

002006' 000000                  VCT:    OPEN
002010' 000000                  LINK:   OPEN
002012' 000000                  SCSR:   OPEN
002014' 000000                  ENDFLG: OPEN
002016' 000110                  TIME:   110
002020' 000000                  TIMER:  0
002022' 000000                  TIME1:  0
002024' 000000                  MASK:   0
002026' 000000                  CHAR:   0

002030' 017000                  TMETBL: 17000
002032' 011700                          11700
002034' 006000                          6000
002036' 005100                          5100
002040' 003000                          3000
002042' 001600                          1600
002044' 000750                          750
```

```
                        350                             ;-150   BPS
                        300                             ;-134.5 BPS
                        260                             ;-110   BPS
                        165                             ;-75    BPS


                .SBTTL MODULE CODE

                ;MODULE INITIALIZATION--THIS ROUTINE INITIALIZES THE DJ11 ASYNCHRONOUS
                ;XMITTER AND RECEIVER, POINTING INTERRUPT VECTORS TO INTERRUPT HANDLERS (VIA A
                ;LINKAGE TABLE) AND INITIALIZING PARAMETERS AND CONTROL REGISTERS
002056' 012767 000101 176034 START:     MOV     #65.,INTR       ;65 INTERRUPTS/ITERATION
002064' 012767 000100 176022           MOV     #64.,WDTO       ;64 WORDS TO MEM/ITERATION
002072' 012767 000100 176016           MOV     #64.,WDFR       ;64 WORDS FROM MEM/ITERATION
002100' 016767 175710 177670 MOV       DVID1,SELECT
002106' 005067 175724                  CLR     ICONT                                   ;MK001
002112' 012704 000306'                 MOV     #XMTTBL,R4
002116' 005024                 1$:     CLR     (R4)+                   ;CLEAR XMT TABLE
002120' 020427 000406'                 CMP     R4,#RCVTBL
002124' 103774                         BLO     1$
002126' 016702 175664                  MOV     SR1,R2                  ;-SET UP TO FIND ITERATION CNT
002132' 042702 000037                  BIC     #37,R2                  ;-STRIP UNWANTED BITS
002136' 012703 002030'                 MOV     #TMETBL,R3              ;-POINT TO ITERATION TABLE
002142' 006302                 2$:     ASL     R2                      ;-SHIFT TO FIND COUNT
002144' 005702                         TST     R2                      ;-IS THIS IT?
002146' 001402                         BEQ     3$                      ;-YES LEAVE NOW
002150' 005723                         TST     (R3)+                   ;-POINT TO NXT IN TBL
002152' 000773                         BR      2$                      ;-TRY AGAIN
002154' 011367 177622          3$:     MOV     (R3),ENDSEL             ;SET UP COUNT
002160' 016767 175630 177604           MOV     DVID1,I                                 ;MK001
002166' 006267 177600          12$:    ASR     I                                       ;MK001
002172' 001403                         BEQ     13$                                     ;MK001
002174' 006267 177602                  ASR     ENDSEL                                  ;MK001
002200' 000772                         BR      12$                                     ;MK001
002202' 012700 000004          13$:    MOV     #4,R0                                   ;MK001
002206' 005001                         CLR     R1
002210' 016761 177570 001712' 4$:      MOV     CNTSEL,RCNT(R1)
002216' 016761 177562 001702'          MOV     CNTSEL,XCNT(R1)
002224' 012761 000001 001722'          MOV     #1,ENDCNT(R1)                           ;MK001
002232' 062701 000002                  ADD     #2,R1
002236' 005300                         DEC     R0
002240' 001363                         BNE     4$
002242' 005767 175570                  TST     ICONT                   ;IS THIS THE FIRST TIME THRU ?
002246' 001031                         BNE     9$                      ;NO BRANCH
002250' 016767 177526 175560           MOV     ENDSEL,ICONT            ;YES SET THE COUNT TO ICONT
002256' 005267 175554                  INC     ICONT                   ;MUST BE GREATER BY ONE BECAUSE
                                                                       ;LAST ENDIT CALL IS DONE AFTER
                                                                       ;A TEST TO SEE IF ALL DEVS
                                                                       ;FUNCTIONED BEFORE WATCHDOG TIMER
                                                                       ;TIMED OUT
002262' 016701 175526                  MOV     DVID1,R1                ;SAVE DVC IN R1
002266' 001313                         BNE     1$                      ;IF SET UP OK - BRANCH
002270' 104410 000000'                 END$,BEGIN              ;
002274' 006201                 11$:    ASR     R1                      ;ANY DEVS LEFT ??
002276' 001707                         BEQ     1$                      ;NO - GOODBYE
```

```
   443  002300' 066767  177476  175530          ADD     ENDSEL,ICONT            ;YUP - SET ADDITIONAL COUNT
   444  002306' 062767  000101  175604          ADD     #65.,INTR               ;65 MORE TRUPS
   445  002314' 062767  000100  175572          ADD     #64.,WDTO               ;64. MORE WORDS TO
   446  002322' 062767  000100  175566          ADD     #64.,WDFR               ;64. MORE WORDS FROM
   447  002330' 000761                          BR      11$                     ;GO CHECK FOR MORE
   448  002332' 012767  100000  177440  9$:     MOV     #100000,EXPLIN          ;SETUP TO PRESET THE RECEIVE TABLE
   449  002340' 012704  000406'                 MOV     #RCVTBL,R4              ;GET START ADDRESS OF RECEIVE TABLE
   450  002344' 016724  177430          5$:     MOV     EXPLIN,(R4)+            ;LOAD RECEIVE TABLE WITH A DONE BIT AND LINE #
   451  002350' 105267  177425                  INCB    EXPLIN+1                ;INCREMENT TO THE NEXT LINE NUMBER
   452  002354' 042767  070000  177416          BIC     #070000,EXPLIN          ;MAKE SURE LINE # NEVER GREATER THAN FIFTEEN
   453  002362' 020427  000606'                 CMP     R4,#SILO                ;SEE IF DONE PRESETTING
   454  002366' 103766                          BLO     5$                      ;BRANCH BACK IF NOT
   455  002370' 012767  177400  177430          MOV     #177400,CHAR            ;-PRESET MASK FOR 8 BIT DATA
   456  002376' 016767  175414  177420          MOV     SR1,MASK                ;-GET OPERATOR INPUT
   457  002404' 042767  177770  177412          BIC     #177770,MASK            ;-ELIMINATE ITERATION BAUD COUNT
   458  002412' 005767  177406                  TST     MASK                    ;8-BIT                          ;MK001
   459  002416' 001405                          BEQ     7$                                                      ;MK001
   460  002420' 006267  177402          6$:     ASR     CHAR                    ;-SHIFT TO RIGHT-
   461  002424' 006267  177374                  ASR     MASK                    ;-SIZE MASK.
   462  002430' 001373                          BNE     6$                      ;-DONE?
   463  002432' 116767  177370  177364  7$:     MOVB    CHAR,MASK               ;-SAVE IT.
   464  002440' 005767  177332          RESTRT: TST     SELECT                  ;ANY DJ'S SELECTED?
   465  002444' 001404                          BEQ     1$                      ;BR IF NO
   466  002446' 032767  177760  177322          BIT     #177760, SELECT         ;UNITS OTHER THAN 0-3 SELECTED?
   467  002454' 001402                          BEQ     2$                      ;NO- OK
   468  002456'                         1$:
   469  002456' 104410  000000'                 END$,BEGIN              ;UNIT SELECTION (DVID1) INCORRECT
   470  002462' 012767  001616' 177272  2$:     MOV     #XQ,XPNTW               ;SETUP QUEUE POINTERS
   471  002470' 012767  001616' 177266          MOV     #XQ,XPNTR
   472  002476' 012767  001626' 177262          MOV     #RQ,RPNTW
   473  002504' 012767  001626' 177256          MOV     #RQ,RPNTR
   474  002512' 012767  001636' 177156          MOV     #ERRQ,ERRQI
   475  002520' 012767  001636' 177152          MOV     #ERRQ,ERRQO
   476  002526' 005067  177262                  CLR     ENDFLG
   477  002532' 012701  000226'                 MOV     #DJLINK,R1              ;GET ADDRESS OF DJ'S LINKING TABLE
   478  002536' 016700  175246                  MOV     VECTOR,R0               ;GET VECTOR ADDRESS
   479  002542' 001403                          BEQ     3$                      ;BR IF VECTOR = ZERO            ;RG03
   480  002544' 016705  175236                  MOV     ADDR,R5                 ;GET CSR ADDRESS
   481  002550' 001005                          BNE     4$                      ;BR IF ADDR NOT = ZERO          ;RG03
   482  002552'                         3$:
   483  002552' 104401  000000' 004461'         MSG$,BEGIN,NOVEC        ;ASCII MESSAGE CALL
   484  002560' 104410  000000'                 END$,BEGIN              ;DROP MODULE            ;RG03
   485  002564' 012767  000001  177202  4$:     MOV     #1,SELMSK
   486  002572' 036767  177176  177176  NXTDEV: BIT     SELMSK,SELECT           ;HAS THIS DEVICE BEEN SELECTED?
   487  002600' 001530                          BEQ     1$                      ;NO- BRANCH
   488  002602' 010120                          MOV     R1,(R0)+                ;YES- LOAD RCV INT VECTOR
   489  002604' 116720  175202                  MOVB    BR1,(R0)+               ;LOAD RCV BR
   490  002610' 105020                          CLRB    (R0)+
   491  002612' 062701  000006                  ADD     #6,R1
   492  002616' 010120                          MOV     R1,(R0)+                ;LOAD XMT INT VECTOR
   493  002620' 116720  175166                  MOVB    BR1,(R0)+               ;LOAD XMT BR
   494  002624' 105020                          CLRB    (R0)+
   495  002626' 062701  000006                  ADD     #6,R1
   496  002632' 004767  001340                  JSR     PC,TBLSNC               ;SYNCH RCV AND XMT TABLES
   497  002636' 012715  000014                  MOV     #14,(R5)                ;CLEAR DJ BUFFER AND UARTS
   498  002642' 012704  000010                  MOV     #10,R4                  ;SETUP WAIT LOOP COUNTER
```

```
                                10$:    BIT     #20,(R5)                ;'BUSY CLEAR'' GONE LOW YET?
                                        BEQ     11$                     ;YES- BRANCH
                                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                        DEC     R4                      ;WAIT LOOP COUNTER
                                        BNE     10$                     ;LOOP IF NOT TIMED OUT
                                        MOV     R1,LINK                 ;SAVE INFO DURING ERROR (CAN BE DONE
                                        MOV     R0,VCT                  ;SINCE THIS IS A SEQUENTIAL LOOP
                                        MOV     R5,SCSR                 ;AND INFO NOT USED ELSEWHERE)
                                        BIC     SELMSK,SELECT           ;CLEAR UNIT SELECTED BIT - DROP UNIT
                                        MOV     R5,CSRA                 ;SETUP FOR ERROR PRINTOUT
                                        MOV     @R5,ACSR
                                        CLR     @R5                     ;DISABLE UNIT
                                        MOV     #16,ERRTYP              ;BUSY STUCK
                                        ;*************************************************************
                                        HRDER$,BEGIN,NULL       ;BUSY CLEAR DIDN'T CLEAR IN TIME
                                        ;*************************************************************
                                        MSGN$,BEGIN,DROPM       ;ASCII MESSAGE CALL WITH COMMON HEADER
                                                                ;'UNIT DROPPED''
                                        MOV     TIME,TIMER              ;RESET TIMER DUE TO PRINTOUT DELAY
                                        TST     SELECT                  ;ANY DJ'S STILL SELECTED?
                                        BNE     12$                     ;YES- BRANCH
                                        END$,BEGIN              ;ALL UNITS DROPPED
                                12$:    MOV     LINK,R1
                                        MOV     VCT,R0
                                        MOV     SCSR,R5
                                        ADD     #10,R5
                                        BR      2$
                                11$:    MOV     #50405,(R5)             ;TURN ON RCVR AND XMIT ENABLE
                                        MOV     R5,R3
                                        SUB     ADDR,R3
                                        ASR     R3
                                        ASR     R3                      ;GET OFFSET (UNIT # TIMES 2)
                                        MOV     XMTSEL(R3),XMITEN(R3)
                                        MOV     XMITEN(R3),XMITSV(R3)   ;SAVE IN CASE OF HANG TO SHOW ACTIVE LINES
                                        MOV     XMITEN(R3),4(R5)        ;TURN ON XMTR
                                        INC     ENDFLG                  ;COUNT UNITS RUNNING
                                        ADD     #10,R5
                                        BR      2$
                                1$:     ADD     #14,R1
                                        ADD     #10,R0
                                        ADD     #10,R5
                                2$:     ASL     SELMSK                  ;NEXT UNIT
                                        CMP     SELMSK,#20              ;DONE?
                                        BLT     NXTDEV                  ;NO- LOOP
                                        MOV     TIME,TIMER
                                        CLR     TIME1                   ;YES- DELAY
                                3$:
                                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                        TST     ENDFLG                  ;DJ11'S DONE?
                                        BNE     4$                      ;NO- BRANCH
                                        ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
                                        JMP     RESTRT                  ;NEXT ITERATION                 ;DRB001
                                4$:     DEC     TIME1                   ;COUNT TIMER
```

```
                        003156' 001362                  BNE     3$                      ;TO CATCH
                        003160' 005367  176634          DEC     TIMER                   ;HUNG DJ11'S
                        003164' 001357                  BNE     3$
                        003166' 104403  000000' 004426' MSGN$,BEGIN,HUNGM      ;ASCII MESSAGE CALL WITH COMMON HEADER
                        003174' 016705  174606          MOV     ADDR,R5
                        003200' 012767  000001  176566  MOV     #1,SELMSK
                        003206' 036767  176562  176562  5$:     BIT     SELMSK,SELECT
                        003214' 001401                  BEQ     6$                      ;BRANCH IF NOT SELECTED
                        003216' 005015                  CLR     @R5                     ;IF SELECTED CLEAR THE CSR
                        003220' 062705  000010          6$:     ADD     #10,R5          ;NEXT CSR ADDRESS
                        003224' 006367  176544          ASL     SELMSK                  ;NEXT UNIT SELECT BIT
                        003230' 026727  176540  000020  CMP     SELMSK,#20              ;ALL UNITS CHECKED?
                        003236' 002763                  BLT     5$                      ;BRANCH IF NOT
                        003240' 104410  000000'         END$,BEGIN              ;DJ11 HUNG

                                                ;ROUTINE TO HANDLE TRANSMITTER INTERRUPTS
                        003244' 010246          XMTINT: MOV     R2,-(SP)                ;SAVE REG 2 ON STACK
                        003246' 011502                  MOV     (R5),R2                 ;GET OFFSET INTO R2
                        003250' 066702  174532          ADD     ADDR,R2                 ;FORM DEVICE ADDRESS
                        003254' 042712  040000          BIC     #040000,(R2)            ;TURN OFF XMT INT ENABLE WHILE YOU SERVICE
                        003260' 011577  176476          MOV     (R5),@XPNTW             ;SAVE OFFSET IN QUEUE
                        003264' 062767  000002  176470  ADD     #2,XPNTW                ;INCREMENT IN POINTER
                        003272' 026727  176464  001626' CMP     XPNTW,#XQ+10            ;TIME TO WRAP?
                        003300' 001003                  BNE     1$                      ;NO
                        003302' 012767  001616' 176452  MOV     #XQ,XPNTW               ;YES
                        003310' 012602          1$:     MOV     (SP)+,R2                ;RESTORE R2
                        003312' 012605                  MOV     (SP)+,R5                ;RESTORE R5
                                                        ;-----------------------------------------------------------------
                        003314' 000004  000000' 003322' PIRQ$,BEGIN,XMTRTN              ; QUEUE UP TO CONTINUE AT XMTRTN AND RTI
                                                        ;-----------------------------------------------------------------


                        003322' 017703  176436  XMTRTN: MOV      @XPNTR,R3              ;GET OFFSET
                        003326' 062767  000002  176430  ADD      #2,XPNTR               ;INCREMENT OUT POINTER
                        003334' 026727  176424  001626' CMP      XPNTR,#XQ+10           ;TIME TO WRAP?
                        003342' 001003                  BNE      1$                     ;NO
                        003344' 012767  001616' 176412  MOV      #XQ,XPNTR              ;YES
                        003352' 010302          1$:     MOV      R3,R2                  ;GET READY- COPY OFFSET
                        003354' 066702  174426          ADD      ADDR,R2                ;TO FORM DEVICE ADDRESS
                        003360' 005712                  TST      (R2)                   ;XMT READY BIT SET?
                        003362' 100422                  BMI      XMTOK                  ;BR IF OK
                        003364' 010267  174510          MOV      R2,CSRA                ;STORE COMMAND REGISTER ADDRESS
                        003370' 011267  174506          MOV      (R2),ACSR              ;STORE CONTENTS OF COMMAND REGISTER
                        003374' 012767  000011  174504  MOV      #11,ERRTYP             ;ILLEGAL INTERRUPT
                                                        ;*****************************************************************
                        003402' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;XMIT READY NOT SET ON XMIT INTERRUPT
                                                        ;*****************************************************************
                        003410' 016767  176402  176402  MOV      TIME,TIMER             ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
                        003416' 052777  040000  174454  BIS      #040000,@CSRA          ;BETTER REENABLE TRANSMITTERS
                        003424' 104400  000000'         EXIT$,BEGIN                    ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                        003430' 010300          XMTOK:  MOV      R3,R0                  ;PREPARE A ONE WORD OFFSET
                        003432' 006200                  ASR      R0                     ;BEGIN THE OFFSET WITH ONE SHIFT
                        003434' 006200                  ASR      R0                     ;COMPLETE THE OFFSET WITH 2ND SHIFT
                        003436' 006303                  ASL      R3                     ;OFFSET OF 20 TIMES UNIT #
                        003440' 116204  000007  MORXMT: MOVB     7(R2),R4               ;PUT LINE # IN R4
                        003444' 042704  177760          BIC      #177760,R4             ;PURGE LEFTOVER DATA FROM R4
```

```
                                        ADD     R3,R4                   ;FORM UNIQUE LINE/CHAR OFFSET
003450' 060304
                                        MOVB    XMTTBL(R4),CHAR         ;-TEMP HLD CHAR
003452' 116467  000306' 176346
                                        BICB    MASK,CHAR               ;-FOR MASKING
003460' 146767  176340  176340
                                        MOVB    CHAR,6(R2)              ;-TRANSMIT CHARACTER
003466' 116762  176334  000006
                                        INCB    XMTTBL(R4)              ;ADVANCE TO NEXT CHARACTER
003474' 105264  000306'
                                        BNE     DECCNT                  ;AVOID NEXT SECTION OF CODE
003500' 001016
                                                                        ;IF SENT CHARS NOT WRAPPED TO ZERO
                                        MOV     #1,R1                   ;BIT POSITION OF THE ONE IS SAME AS XMT LINE #
003502' 012701  000001
                                        BIC     #177760,R4              ;LEAVE ONLY LINE # IN R4
003506' 042704  177760
                                        BEQ     LINOFF                  ;LINE 0 IS DONE,TURN IT OFF
003512' 001403
                                1$:     ASL     R1                      ;SHIFT XMT BIT TO NEXT XMTTER LINE POSITION
003514' 006301
                                        DEC     R4                      ;DECREMENT ACTUAL LINE NUMBER
003516' 005304
                                        BNE     1$                      ;BRANCH BACK IF NOT DONE
003520' 001375
                                LINOFF: BIC     R1,XMITEN(R0)           ;TURN OFF THE INDIVIDUAL  XMTTER LINE
003522' 040160  001742'
                                        BNE     DECCNT                  ;ARE ALL LINES DONE NOW?
003526' 001003
                                        MOV     XMTSEL(R0),XMITEN(R0)   ;IF YES, RESET ENABLE WORD
003530' 016060  001732' 001742'
                                DECCNT: DEC     XCNT(R0)                ;DECREMENT CHARACTER COUNT
003536' 005360  001702'
                                        BLE     OFF                     ;HAVE YOU SENT 65? BR IF YES
003542' 003403
                                        TST     (R2)                    ;ADDITIONAL XMTTRS READY
003544' 005712
                                        BMI     MORXMT                  ;BR IF YES
003546' 100734
                                        BR      XMTOUT                  ;GO OUT AND WAIT FOR ANOTHER LINE TO INT
003550' 000407
                                OFF:    CLR     4(R2)                   ;TURN OFF XMTTER
003552' 005062  000004
                                        MOV     CNTSEL,XCNT(R0)         ;RESET CHAR COUNT
003556' 016760  176222  001702'
                                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003564' 104400  000000'
                                XMTOUT: BIS     #040000,(R2)            ;ENABLE MORE XMITR INTERRUPTS
003570' 052712  040000
                                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003574' 104400  000000'

                                ;ROUTINE TO HANDLE RECEIVER INTERRUPTS
                                RCVINT: MOV     R2,-(SP)                ;SAVE R2 ON STACK
003600' 010246
                                        MOV     (R5),R2                 ;GET OFFSET
003602' 011502
                                        ADD     ADDR,R2                 ;FORM DEVICE ADDRESS
003604' 066702  174176
                                        BIC     #010000,(R2)            ;TURN OFF INTERRUPTS
003610' 042712  010000
                                        MOV     (R5),@RPNTW             ;SAVE OFFSET IN QUEUE
003614' 011577  176146
                                        ADD     #2,RPNTW                ;INCREMENT WRITE POINTER
003620' 062767  000002  176140
                                        CMP     RPNTW,#RQ+10            ;TIME TO WRAP?
003626' 026727  176134  001636'
                                        BNE     1$                      ;NO
003634' 001003
                                        MOV     #RQ,RPNTW               ;YES
003636' 012767  001626' 176122
                                1$:     MOV     (SP)+,R2                ;RESTORE R2
003644' 012602
                                        MOV     (SP)+,R5                ;RESTORE R5
003646' 012605
                                        ;---------------------------------------------------------------
003650' 000004  000000' 003656'         PIRQ$,BEGIN,RCVRTN              ; QUEUE UP TO CONTINUE AT RCVRTN AND RTI
                                        ;---------------------------------------------------------------

                                RCVRTN: MOV     @RPNTR,R3               ;GET OFFSET
003656' 017703  176106
                                        ADD     #2,RPNTR                ;INCREMENT READ POINTER
003662' 062767  000002  176100
                                        CMP     RPNTR,#RQ+10            ;TIME TO WRAP?
003670' 026727  176074  001636'
                                        BNE     1$                      ;NO
003676' 001003
                                        MOV     #RQ,RPNTR               ;YES
003700' 012767  001626' 176062
                                1$:     MOV     R3,R0                   ;GET OFFSET AGAIN
003706' 010300
                                        ASR     R0                      ;FORM A ONE WORD
003710' 006200
                                        ASR     R0                      ;INDEX
003712' 006200
                                PASSED: MOV     R3,R2                   ;GET READY
003714' 010302
                                        ADD     ADDR,R2                 ;TO FORM DEVICE ADDRESS
003716' 066702  174064
                                        BIT     #020000,(R2)            ;SEE IF BUFFER SILO FULL SET
003722' 032712  020000
                                        BNE     CONT1                   ;CONTINUE IF OK
003726' 001022
                                        MOV     R2,CSRA                 ;STORE COMMAND REGISTER ADDRESS
003730' 010267  174144
```

```
                        MOV     (R2),ACSR               ;STORE CONTENTS OF COMMAND REGISTER
                        MOV     #11,ERRTYP              ;ILLEGAL INTERRUPT
                        ;*************************************************************
                        HRDER$,BEGIN,NULL       ;FALSE INTERRUPT
                        ;*************************************************************
                        MOV     TIME,TIMER              ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
                        BIS     #010000,@CSRA           ;BETTER ENABLE RECEIVER BEFORE LEAVING
                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

CONT1:                  MOV     R3,R4                   ;BEGIN TO FORM
                        ASL     R4                      ;A NEW OFFSET
                        ASL     R4                      ;INTO EXPECTED RECEIVE DATA TABLE
                        MOV     R4,R5                   ;BETTER SAVE IT
                        ASL     R4                      ;SO YOU CAN FORM A SILO OFFSET
                        ASL     R4
                        ADD     R0,R4                   ;OFFSET = 202(OCTAL) TIMES UNIT#
                        ADD     #SILO,R4                ;FORM SILO BASE OFFSET
                        MOV     R4,R1                   ;REMEMBER FIRST SILO WORD ADDRESS
NOTDUN:                 MOV     2(R2),(R4)+             ;MOVE DATA FROM DJ TO SILO BUFFER
                        DEC     RCNT(R0)                ;DECREMENT # OF CHARACTERS RECEIVED
                        BGT     NOTDUN                  ;BR IF NOT RECEIVED 65 YET
                        MOV     CNTSEL,RCNT(R0)         ;RESET RECEIVE COUNT
                        ADD     #RCVTBL,R5              ;FORM OFFSET INTO EXP RECV DATA TABLE
                        MOV     R5,I                    ;SAVE IT IN 'I' AS AN INDEX
                                                        ;(RCVTBL BASE ADDRESS FOR THIS UNIT)
NXTCHR:                 MOVB    1(R1),R5                ;GET LINE NUMBER FROM DATA WORD
                        BIC     #177760,R5              ;MASK SO HAVE ONLY LINE #
                        ASL     R5                      ;MAKE LINE# EVEN WORD BOUNDARY
                        ADD     I,R5                    ;FORM EXPECTED DATA ADDRESS
                        MOV     (R5),CHAR               ;-TEMP HLDING FOR BIT SIZING
                        BICB    MASK,CHAR               ;-CHECK ONLY STRAPPED BITS
                        CMP     CHAR,(R1)+              ;-COMPARE EXPECTED VS ACTUAL DATA
                        BNE     DATERR                  ;BR IF BAD DATA
DONE:                   INCB    (R5)                    ;INCREMENT EXPECTED DATA
                        CMP     R1,R4                   ;HAVE YOU FINISHED ENTIRE BUFFER?
                        BLO     NXTCHR                  ;GO BACK IF NOT
                        DEC     ENDCNT(R0)
                        BGT     1$
                        MOV     #1,ENDCNT(R0)                                   ;MK001
                        BIC     #050405,(R2)            ;TURN OFF DJ11
                        DEC     ENDFLG
                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
1$:                     MOV     R2,R5                   ;COPY CSR ADDRESS TO R5
                        JSR     PC,TBLSNC               ;SYNCH RCV AND XMT TABLES
                        BIS     #50000,(R2)             ;NO- ENABLE SILO FULL INT AND XMTR INT
                        MOV     XMITEN(R0),4(R2)        ;TURN XMITTERS BACK ON
                        MOV     XMITEN(R0),XMITSV(R0)   ;SAVE INDICATOR OF LINES ACTIVATED
                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                        ;SYNCHRONIZE RECEIVER AND TRANSMITTER TABLES FOR ONE UNIT
                        ;WHOSE CSR ADDRESS IS IN R5
TBLSNC:                 MOV     R5,-(SP)                ;CSR ADDRESS
                        SUB     ADDR,(SP)
                        ASL     (SP)                    ;GET OFFSET (UNIT # TIMES 20 OCTAL)
                        MOV     (SP),-(SP)              ;COPY OFFSET
                        ASL     (SP)                    ;OFFSET INTO RCV TABLE (UNIT # TIMES 40)
```

```
723  004212' 062716  000406'                ADD     #RCVTBL,(SP)
724  004216' 062766  000306' 000002          ADD     #XMTTBL,2(SP)
725  004224' 012746  000020                   MOV     #16.,-(SP)              ;COUNTER
726  004230' 117676  000004  000002  1$:      MOVB    @4(SP),@2(SP)           ;MOVE XMIT TABLE CONTENTS INTO DATA
727                                                                          ;PORTION OF RCV TABLE
728
729  004236' 005266  000004                   INC     4(SP)
730  004242' 062766  000002  000002           ADD     #2,2(SP)
731  004250' 005316                           DEC     (SP)
732  004252' 001366                           BNE     1$
733  004254' 022626                           CMP     (SP)+,(SP)+             ;FIX SP
734  004256' 005726                           TST     (SP)+
735  004260' 000207                           RTS     PC
736
737                                   ;ERROR HANDLER FOR DATA ERRORS
738  004262' 010267  173612           DATERR: MOV     R2,CSRA                 ;STORE COMMAND REGISTER ADDRESS
739  004266' 010567  173610                   MOV     R5,SBADR                ;STORE ADDRESS OF TEST DATA
740  004272' 005741                           TST     -(R1)                   ;RESET R1 BACK TO THE ADDRESS OF THE FAIL DATA
741  004274' 010167  173604                   MOV     R1,WASADR               ;STORE ADDRESS OF XMTTED DATA
742  004300' 011567  173602                   MOV     (R5),ASB                ;STORE EXPECTED RECEIVE DATA
743  004304' 012167  173600                   MOV     (R1)+,AWAS              ;STORE ACTUAL RECEIVED DATA AND RESET R1 TO RIGH
744  004310' 016702  175362                   MOV     ERRQI,R2
745  004314' 146767  175504  173564           BICB    MASK,ASB                ;-MASK OUT STRAPPED OFF BITS
746  004322' 010022                           MOV     R0,(R2)+
747  004324' 010122                           MOV     R1,(R2)+
748  004326' 010422                           MOV     R4,(R2)+
749  004330' 010522                           MOV     R5,(R2)+
750  004332' 020227  001676'                  CMP     R2,#ERRQ+32.
751  004336' 103402                           BLO     1$
752  004340' 012702  001636'                  MOV     #ERRQ,R2
753  004344' 010267  175326           1$:     MOV     R2,ERRQI
754                                           ;*************************************************************************
755  004350' 104404  000000'                  DATER$,BEGIN                    ;DATA ERROR!!!
756                                           ;*************************************************************************
757  004354' 016767  175436  175436           MOV     TIME,TIMER              ;RESET TIMER DUE TO PRINTOUT DELAY
758  004362' 016702  175312                   MOV     ERRQO,R2
759  004366' 012200                           MOV     (R2)+,R0
760  004370' 012201                           MOV     (R2)+,R1
761  004372' 012204                           MOV     (R2)+,R4
762  004374' 012205                           MOV     (R2)+,R5
763  004376' 020227  001676'                  CMP     R2,#ERRQ+32.
764  004402' 103402                           BLO     2$
765  004404' 012702  001636'                  MOV     #ERRQ,R2
766  004410' 010267  175264           2$:     MOV     R2,ERRQO
767  004414' 016702  173460                   MOV     CSRA,R2
768  004420' 000632                           BR      DONE                    ;NOW CONTINUE CHECKING DATA
769
770  004422' 004432'                  DROPM:  DROP
771  004424' 177777                           -1
772  004426' 004447'                  HUNGM:  HUNG
773  004430' 177777                           -1
774
775  004432' 047125  052111  042040  DROP:    .ASCIZ  /UNIT DROPPED/
776  004440' 047522  050120  042105
777  004446'    000
778  004447'    125  044516  020124  HUNG:    .ASCIZ  /UNIT HUNG/
```

```
   779  004454' 052510  043516      000
   780  004461'    045  045104  020101  NOVEC:  .ASCIZ  /%DJA MODULE: VECTOR + ADDR  LOCS. HAVE NOT BEEN LOADED.%/
   781  004466' 047515  052504  042514
   782  004474' 020072  042526  052103
   783  004502' 051117  025440  040440
   784  004510' 042104  020122  046040
   785  004516' 041517  027123  044040
   786  004524' 053101  020105  047516
   787  004532' 020124  042502  047105
   788  004540' 046040  040517  042504
   789  004546' 027104  000045
   790                             .EVEN
   791          000001             .END
```

```
ACSR     000102R          263#    510*    596*    667*
ADDR     000006R          229#    480     529     559     573     592     641     663     719
ADDR22=  001000           281#
APTPRE=  000200           281#
ASB      000106R          267#    742*    745*
ASTAT    000104R          265#
AUTO  =  000010           281#
AWAS     000110R          268#    743*
BEGIN    000000R          226#    440     469     483     484     501     502     514     516     521     547     548     551
                          558     568     583     599     603     634     636     651     670     674     708     714     755
BIT0  =  000001           281#
BIT1  =  000002           281#
BIT10 =  002000           281#
BIT11 =  004000           281#
BIT12 =  010000           281#
BIT13 =  020000           281#
BIT14 =  040000           281#
BIT15 =  100000           281#
BIT2  =  000004           281#
BIT3  =  000010           281#
BIT4  =  000020           281#
BIT5  =  000040           281#
BIT6  =  000100           281#
BIT7  =  000200           281#
BIT8  =  000400           281#
BIT9  =  001000           281#
BREAK$=  104407           281#    501     502     547     548
BR1      000012R          231#    489     493
BR2      000013R          232#
BTOD$ =  104421           281#
CAPRES=  000004           281#
CDATA$=  104412           281#
CHAR     002026R          378#    455*    460*    463     612*    613*    614     696*    697*    698
CKHNG$=  000001           281#
CLKPRE=  000001           281#
CLKSP$=  104422           281#
CNTSEL   002004R          368#    424     425     633     688
CONFIG   000056R          251#
CONT1    003774R          665     676#
CSRA     000100R          261#    509*    595*    602*    666*    673*    738*    767
DATCK$=  104411           281#
DATERR   004262R          699     738#
DATER$=  104404           281#    755
DECCNT   003536R          616     625     627#
DJLINK   000226R          294#    477
DONE     004106R          700#    768
DROP     004432R          770     775#
DROPM    004422R          516     770#
DVID1    000014R          233#    402     417     438
ECCMEM=  000100           281#
ENDCNT   001722R          337#    426*    703*    705*
ENDFLG   002014R          373#    476*    535*    549     707*
ENDIT$=  104413           281#    551
ENDSEL   002002R          367#    416*    420*    432     443
END$  =  104410           281#    440     469     484     521     568
ERRQ     001636R          323#    474     475     750     752     763     765
```

```
ERRQI   001676R          324#    474*    744     753*
ERRQO   001700R          325#    475*    758     766*
ERRTYP  000106R          266#    512*    597*    668*
EXIT$ = 104400           281#    603     634     636     674     708     714
EXPLIN  002000R          365#    448*    450     451*    452*
GETPA$= 104415           281#
GWBUF$= 104414           281#
HRDCNT  000044R          246#
HRDER$= 104405           281#    514     599     670
HRDPAS  000050R          248#
HUNG    004447R          772     778#
HUNGM   004426R          558     772#
I       001772R          362#    417*    418*    690*    695
ICONT   000036R          243#    403*    430     432*    433*    443*
ICOUNT  000040R          244#
IDNUM   000122R          273#
INDPAR= 000040           281#
INIT    000030R          240#
INTR    000120R          272#    399*    444*
KTPRES= 000400           281#
KTXTND= 040000           281#
LINK    002010R          371#    505*    522
LINOFF  003522R          620     624#
MAP22$= 104416           281#
MASK    002024R          377#    456*    457*    458     461*    463*    613     697     745
MODNAM  000000R          227#
MODREV  000224R          289#
MODSP   000224R          241     279#
MORXMT  003440R          609#    630
MSGN$ = 104403           281#    516     558
MSGS$ = 104402           281#
MSG$  = 104401           281#    483
NCPUOP= 000020           281#
NOAPTY= 000002           281#
NOTDUN  004020R          685#    687
NOVEC   004461R          483     780#
NULL  = 000000           281#    514     599     670
NXTCHR  004050R          692#    702
NXTDEV  002572R          486#    543
OFF     003552R          628     632#
OPEN  = 000000           228     234     235     236     237     254     255     256     257     258     259     260     261
                         263     265     267     268     270     271     272     281#    311#    324     325     370     371
                         372     373
OTOA$ = 104420           281#
PARPRE= 002000           281#
PASCNT  000034R          242#
PASSED  003714R          662#
PDPLSI= 020000           281#
PDP23 = 000002           281#
PDP24 = 000004           281#
PDP44 = 100000           281#
PDP60 = 004000           281#
PDP70 = 010000           281#
PIRQ$ = 000004           281#    583     651
POPSP = 005726           281#
POPSP2= 022626           281#
```

```
PRHMS$= 000002           281#
PRTY  = 000000           232     281#
PRTY0 = 000000           281#
PRTY1 = 000040           281#
PRTY2 = 000100           281#
PRTY3 = 000140           281#
PRTY4 = 000200           281#
PRTY5 = 000240           231     281#
PRTY6 = 000300           281#
PRTY7 = 000340           281#
PS    = 177776           281#
PSW   = 177776           281#
PUSH  = 005746           281#
PUSH2 = 024646           281#
PWRFLG= 000002           281#
RAND$ = 104417           281#
RANNUM  000054R          250#
RCNT    001712R          332#    424*    686*    688*
RCVINT  003600R          294     298     302     306     639#
RCVRTN  003656R          651     654#
RCVTBL  000406R          315#    406     449     689     723
RESTRT  002440R          269     464#    553
RES1    000056R          252#
RES2    000060R          253#
RH70  = 001000           281#
RPNTR   001770R          360#    473*    654     655*    656     658*
RPNTW   001766R          359#    472*    643*    644*    645     647*
RQ      001626R          321#    359     360     472     473     645     647     656     658
RSTRT   000112R          269#
SBADR   000102R          262#    739*
SCSR    002012R          372#    507*    524
SELECT  001776R          364#    402*    464     466     486     508*    519     561
SELMSK  001774R          363#    485*    486     508     541*    542     560*    561     565*    566
SILO    000606R          317#    453     683
SOFCNT  000042R          245#
SOFER$= 104406           281#
SOFPAS  000046R          247#
SPOINT  000032R          241#
SPSIZ = 000040             1#    274
SR1     000016R          234#    408     456
SR2     000020R          235#
SR3     000022R          236#
SR4     000024R          237#
START   002056R          240     399#
STAT    000026R          239#
SVR0    000062R          254#
SVR1    000064R          255#
SVR2    000066R          256#
SVR3    000070R          257#
SVR4    000072R          258#
SVR5    000074R          259#
SVR6    000076R          260#
SYSCNT  000052R          249#
TBLSNC  004176R          496     710     718#
TIME    002016R          374#    518     544     601     672     757
TIMER   002020R          375#    518*    544*    556*    601*    672*    757*
```

```
TIME1   002022R          376#    545*    554*
TMETBL  002030R          380#    410
TRPDFD= 000023           281#
USTACK= 000001           281#
VCT     002006R          370#    506*    523
VECTOR  000010R          230#    478
WASADR  000104R          264#    741*
WDFR    000116R          271#    401*    446*
WDTO    000114R          270#    400*    445*
XCNT    001702R          327#    425*    627*    633*
XFLAG   000005R          228#
XMITEN  001742R          344#    532*    533     534     624*    626*    712     713
XMITSV  001752R          349#    533*    713*
XMTINT  003244R          296     300     304     308     571#
XMTOK   003430R          594     605#
XMTOUT  003570R          631     635#
XMTRTN  003322R          583     586#
XMTSEL  001732R          341#    532     626
XMTTBL  000306R          313#    404     612     615*    724
XPNTR   001764R          358#    471*    586     587*    588     590*
XPNTW   001762R          357#    470*    575*    576*    577     579*
XQ      001616R          319#    357     358     470     471     577     579     588     590
.     = 004552R          313#    315#    317#    319#    321#    323#

. ABS.  000000     000
        004552     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

DE2:XDJAL0,DE2:XDJAL0.SEQ/NL:TOC/SOL/CRF/DOC=DDXCOM,XDJAL0
RUN-TIME:  1  2  .3 SECONDS
RUN-TIME RATIO:  12/4=2.5
CORE USED:  10K  (20 PAGES)

DOCUMENT PAGES:  20
```