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:
    1-bit black&white, compressed for minimal file size

```
                .REM %
```

IDENTIFICATION

PRODUCT CODE: AC-E721L-MC

PRODUCT NAME: CXDJALO 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     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
REPOPTED 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 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
```

```
001712' 000101  000101  000101  RCNT:.WORD 101,101,101,101      ;NUMBER OF CHARACTERS TO BE PROCESSED BY THE
001720' 000101                                                  ;RECEIVE ROUTINE. A WORD FOR EACH DEVICE

001722' 000200  000200  000200  ENDCNT:.WORD 200,200,200,200          ;NUMBER OF ''65 CHARACTER CHUNKS''
001730' 000200                                                        ;TO BE PROCESSED BY THE MODULE PER PASS

001732' 177777  177777  177777  XMTSEL: .WORD   177777,177777,177777,177777     ;LINES TO BE RUN
001740' 177777

001742' 177777  177777  177777  XMITEN:.WORD 177777,177777,177777,177777        ;WORD USED TO FILL TRANSMITTER COMMAND
001750' 177777                                                                  ;REGISTER 16XXX2,CORRESPONDS TO
                                                                                ;A ONE FOR EACH ACTIVE LINE

001752' 000000  000000  000000  XMITSV: .WORD 0,0,0,0                  ;SAVE LINES ACTIVATED IN CASE OF HANG
001760' 000000


001762' 001616'                 XPNTW:  XQ                      ;POINTERS USED FOR THE BOOKKEEPING OF
001764' 001616'                 XPNTR:  XQ                      ;THE TRANSMIT QUEUE
001766' 001626'                 RPNTW:  RQ                      ;POINTERS USED FOR THE BOOKKEEPING OF
001770' 001626'                 RPNTR:  RQ                      ;THE RECEIVE QUEUE

001772' 000000                  I:      000                     ;TEMPORARY STORAGE LOCATION.USED DURING RECEIVE DATA CHE
001774' 000001                  SELMSK: 1                       ;MASK TO CHECK FOR UNIT SELECTED
001776' 000000                  SELECT: 0
002000' 100000                  EXPLIN: 100000                  ;USED DURING INITIALIZATION,THE ONE REPRESENTS THE
                                                                ;DONE BIT IN THE EXPECTED RECEIVE DATA TABLE
002002' 000200                  ENDSEL: 200                     ;REFILL OF 'ENDSEL' WORD AFTER ENDPASS
002004' 000101                  CNTSEL: 101                     ;REFILL OF 'RCNT' AND 'XCNT' AFTER AN INTERRUPT SERVICE
                                                                ;NOT VARIABLE SINCE 65. CHARS REQD TO GET SILO FULL INT
002006' 000000                  VCT:    OPEN                    ;HOLDS VECTOR ADDRESS DURING SETUP ERROR PRINTOUT
002010' 000000                  LINK:   OPEN                    ;HOLDS LINK ADDRESS DURING SETUP ERROR PRINTOUT
002012' 000000                  SCSR:   OPEN
002014' 000000                  ENDFLG: OPEN                    ;NUMBER OF DJ11'S STILL RUNNING
002016' 000110                  TIME:   110                     ;WATCHDOG TIMER OUTER LOOP VALUE
002020' 000000                  TIMER:  0
002022' 000000                  TIME1:  0
002024' 000000                  MASK:   0                               ;-DATA BIT LENGTH (5-8) MASK
002026' 000000                  CHAR:   0                               ;-TEMP HOLDING FOR MASK & ITERATION COUNT &
                                                                        ;-DATA COMPARSION DURRING TEST.
002030' 017000                  TMETBL: 17000                           ;-9600  BPS VARIABLE
002032' 011700                          11700                           ;-4800  BPS ITERATION
002034' 006000                          6000                            ;-2400  BPS COUNTS
002036' 005100                          5100                            ;-1800  BPS PER
002040' 003000                          3000                            ;-1200  BPS LINE
002042' 001600                          1600                            ;-600   BPS SPEEDS
002044' 000750                          750                             ;-300   BPS
```

```
                        000350                          ;-150   BPS
                        000300                          ;-134.5 BPS
                        000260                          ;-110   BPS
                        000165                          ;-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
```

```
                                ADD     ENDSEL,ICONT            ;YUP - SET ADDITIONAL COUNT
                                ADD     #65.,INTR               ;65 MORE TRUPS
                                ADD     #64.,WDTO               ;64. MORE WORDS TO
                                ADD     #64.,WDFR               ;64. MORE WORDS FROM
                                BR      11$                     ;GO CHECK FOR MORE
                        9$:     MOV     #100000,EXPLIN          ;SETUP TO PRESET THE RECEIVE TABLE
                                MOV     #RCVTBL,R4              ;GET START ADDRESS OF RECEIVE TABLE
                        5$:     MOV     EXPLIN,(R4)+            ;LOAD RECEIVE TABLE WITH A DONE BIT AND LINE #
                                INCB    EXPLIN+1                ;INCREMENT TO THE NEXT LINE NUMBER
                                BIC     #070000,EXPLIN          ;MAKE SURE LINE # NEVER GREATER THAN FIFTEEN
                                CMP     R4,#SILO                ;SEE IF DONE PRESETTING
                                BLO     5$                      ;BRANCH BACK IF NOT
                                MOV     #177400,CHAR            ;-PRESET MASK FOR 8 BIT DATA
                                MOV     SR1,MASK                ;-GET OPERATOR INPUT
                                BIC     #177770,MASK            ;-ELIMINATE ITERATION BAUD COUNT
                                TST     MASK                    ;8-BIT                          ;MK001
                                BEQ     7$                                                      ;MK001
                        6$:     ASR     CHAR                    ;-SHIFT TO RIGHT-
                                ASR     MASK                    ;-SIZE MASK.
                                BNE     6$                      ;-DONE?
                        7$:     MOVB    CHAR,MASK               ;-SAVE IT.
                        RESTRT: TST     SELECT                  ;ANY DJ'S SELECTED?
                                BEQ     1$                      ;BR IF NO
                                BIT     #177760, SELECT         ;UNITS OTHER THAN 0-3 SELECTED?
                                BEQ     2$                      ;NO- OK
                        1$:
                                END$,BEGIN              ;UNIT SELECTION (DVID1) INCORRECT
                        2$:     MOV     #XQ,XPNTW               ;SETUP QUEUE POINTERS
                                MOV     #XQ,XPNTR
                                MOV     #RQ,RPNTW
                                MOV     #RQ,RPNTR
                                MOV     #ERRQ,ERRQI
                                MOV     #ERRQ,ERRQO
                                CLR     ENDFLG
                                MOV     #DJLINK,R1              ;GET ADDRESS OF DJ'S LINKING TABLE
                                MOV     VECTOR,R0               ;GET VECTOR ADDRESS
                                BEQ     3$                      ;BR IF VECTOR = ZERO            ;RG03
                                MOV     ADDR,R5                 ;GET CSR ADDRESS
                                BNE     4$                      ;BR IF ADDR NOT = ZERO          ;RG03
                        3$:
                                MSG$,BEGIN,NOVEC        ;ASCII MESSAGE CALL
                                END$,BEGIN              ;DROP MODULE            ;RG03
                        4$:     MOV     #1,SELMSK
                        NXTDEV: BIT     SELMSK,SELECT           ;HAS THIS DEVICE BEEN SELECTED?
                                BEQ     1$                      ;NO- BRANCH
                                MOV     R1,(R0)+                ;YES- LOAD RCV INT VECTOR
                                MOVB    BR1,(R0)+               ;LOAD RCV BR
                                CLRB    (R0)+
                                ADD     #6,R1
                                MOV     R1,(R0)+                ;LOAD XMT INT VECTOR
                                MOVB    BR1,(R0)+               ;LOAD XMT BR
                                CLRB    (R0)+
                                ADD     #6,R1
                                JSR     PC,TBLSNC               ;SYNCH RCV AND XMT TABLES
                                MOV     #14,(R5)                ;CLEAR DJ BUFFER AND UARTS
                                MOV     #10,R4                  ;SETUP WAIT LOOP COUNTER
```

```
   499  002646' 032715  000020          10$:    BIT     #20,(R5)                ;'BUSY CLEAR'' GONE LOW YET?
   500  002652' 001456                          BEQ     11$                     ;YES- BRANCH
   501  002654' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
   502  002660' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
   503  002664' 005304                          DEC     R4                      ;WAIT LOOP COUNTER
   504  002666' 001367                          BNE     10$                     ;LOOP IF NOT TIMED OUT
   505  002670' 010167  177114                  MOV     R1,LINK                 ;SAVE INFO DURING ERROR (CAN BE DONE
   506  002674' 010067  177106                  MOV     R0,VCT                  ;SINCE THIS IS A SEQUENTIAL LOOP
   507  002700' 010567  177106                  MOV     R5,SCSR                 ;AND INFO NOT USED ELSEWHERE)
   508  002704' 046767  177064  177064          BIC     SELMSK,SELECT           ;CLEAR UNIT SELECTED BIT - DROP UNIT
   509  002712' 010567  175162                  MOV     R5,CSRA                 ;SETUP FOR ERROR PRINTOUT
   510  002716' 011567  175160                  MOV     @R5,ACSR
   511  002722' 005015                          CLR     @R5                     ;DISABLE UNIT
   512  002724' 012767  000016  175154          MOV     #16,ERRTYP              ;BUSY STUCK
   513                                          ;*****************************************************************
   514  002732' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BUSY CLEAR DIDN'T CLEAR IN TIME
   515                                          ;*****************************************************************
   516  002740' 104403  000000' 004422'         MSGN$,BEGIN,DROPM       ;ASCII MESSAGE CALL WITH COMMON HEADER
   517                                                                  ;'UNIT DROPPED''
   518  002746' 016767  177044  177044          MOV     TIME,TIMER              ;RESET TIMER DUE TO PRINTOUT DELAY
   519  002754' 005767  177016                  TST     SELECT                  ;ANY DJ'S STILL SELECTED?
   520  002760' 001002                          BNE     12$                     ;YES- BRANCH
   521  002762' 104410  000000'                 END$,BEGIN              ;ALL UNITS DROPPED
   522  002766' 016701  177016          12$:    MOV     LINK,R1
   523  002772' 016700  177010                  MOV     VCT,R0
   524  002776' 016705  177010                  MOV     SCSR,R5
   525  003002' 062705  000010                  ADD     #10,R5
   526  003006' 000433                          BR      2$
   527  003010' 012715  050405          11$:    MOV     #50405,(R5)             ;TURN ON RCVR AND XMIT ENABLE
     8  003014' 010503                          MOV     R5,R3
   529  003016' 166703  174764                  SUB     ADDR,R3
   530  003022' 006203                          ASR     R3
   531  003024' 006203                          ASR     R3                      ;GET OFFSET (UNIT # TIMES 2)
   532  003026' 016363  001732' 001742'         MOV     XMTSEL(R3),XMITEN(R3)
   533  003034' 016363  001742' 001752'         MOV     XMITEN(R3),XMITSV(R3)   ;SAVE IN CASE OF HANG TO SHOW ACTIVE LINES
   534  003042' 016365  001742' 000004          MOV     XMITEN(R3),4(R5)        ;TURN ON XMTR
   535  003050' 005267  176740                  INC     ENDFLG                  ;COUNT UNITS RUNNING
   536  003054' 062705  000010                  ADD     #10,R5
   537  003060' 000406                          BR      2$
   538  003062' 062701  000014          1$:     ADD     #14,R1
   539  003066' 062700  000010                  ADD     #10,R0
   540  003072' 062705  000010                  ADD     #10,R5
   541  003076' 006367  176672          2$:     ASL     SELMSK                  ;NEXT UNIT
   542  003102' 026727  176666  000020          CMP     SELMSK,#20              ;DONE?
   543  003110' 002630                          BLT     NXTDEV                  ;NO- LOOP
   544  003112' 016767  176700  176700          MOV     TIME,TIMER
   545  003120' 005067  176676                  CLR     TIME1                   ;YES- DELAY
   546  003124'                         3$:
   547  003124' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
   548  003130' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
   549  003134' 005767  176654                  TST     ENDFLG                  ;DJ11'S DONE?
   550  003140' 001004                          BNE     4$                      ;NO- BRANCH
   551  003142' '04413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
   552                                                                  ;MONITOR SHALL TEST END OF PASS
   553  003146' 000167  177266                  JMP     RESTRT                  ;NEXT ITERATION                 ;DRB001
   554  003152' 005367  176644          4$:     DEC     TIME1                   ;COUNT TIMER
```

```
                        BNE     3$                      ;TO CATCH
                        DEC     TIMER                   ;HUNG DJ11'S
                        BNE     3$
                        MSGN$,BEGIN,HUNGM       ;ASCII MESSAGE CALL WITH COMMON HEADER
                        MOV     ADDR,R5
                        MOV     #1,SELMSK
5$:                     BIT     SELMSK,SELECT
                        BEQ     6$                      ;BRANCH IF NOT SELECTED
                        CLR     @R5                     ;IF SELECTED CLEAR THE CSR
6$:                     ADD     #10,R5                  ;NEXT CSR ADDRESS
                        ASL     SELMSK                  ;NEXT UNIT SELECT BIT
                        CMP     SELMSK,#20              ;ALL UNITS CHECKED?
                        BLT     5$                      ;BRANCH IF NOT
                        END$,BEGIN              ;DJ11 HUNG

                ;ROUTINE TO HANDLE TRANSMITTER INTERRUPTS
XMTINT:         MOV     R2,-(SP)                ;SAVE REG 2 ON STACK
                MOV     (R5),R2                 ;GET OFFSET INTO R2
                ADD     ADDR,R2                 ;FORM DEVICE ADDRESS
                BIC     #040000,(R2)            ;TURN OFF XMT INT ENABLE WHILE YOU SERVICE
                MOV     (R5),@XPNTW             ;SAVE OFFSET IN QUEUE
                ADD     #2,XPNTW                ;INCREMENT IN POINTER
                CMP     XPNTW,#XQ+10            ;TIME TO WRAP?
                BNE     1$                      ;NO
                MOV     #XQ,XPNTW               ;YES
1$:             MOV     (SP)+,R2                ;RESTORE R2
                MOV     (SP)+,R5                ;RESTORE R5
                ;-----------------------------------------------------------------
                PIRQ$,BEGIN,XMTRTN              ; QUEUE UP TO CONTINUE AT XMTRTN AND RTI
                ;-----------------------------------------------------------------


XMTRTN:         MOV     @XPNTR,R3               ;GET OFFSET
                ADD     #2,XPNTR                ;INCREMENT OUT POINTER
                CMP     XPNTR,#XQ+10            ;TIME TO WRAP?
                BNE     1$                      ;NO
                MOV     #XQ,XPNTR               ;YES
1$:             MOV     R3,R2                   ;GET READY- COPY OFFSET
                ADD     ADDR,R2                 ;TO FORM DEVICE ADDRESS
                TST     (R2)                    ;XMT READY BIT SET?
                BMI     XMTOK                   ;BR IF OK
                MOV     R2,CSRA                 ;STORE COMMAND REGISTER ADDRESS
                MOV     (R2),ACSR               ;STORE CONTENTS OF COMMAND REGISTER
                MOV     #11,ERRTYP              ;ILLEGAL INTERRUPT
                ;*****************************************************************
                HRDER$,BEGIN,NULL       ;XMIT READY NOT SET ON XMIT INTERRUPT
                ;*****************************************************************
                MOV     TIME,TIMER              ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
                BIS     #040000,@CSRA           ;BETTER REENABLE TRANSMITTERS
                EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

XMTOK:          MOV     R3,R0                   ;PREPARE A ONE WORD OFFSET
                ASR     R0                      ;BEGIN THE OFFSET WITH ONE SHIFT
                ASR     R0                      ;COMPLETE THE OFFSET WITH 2ND SHIFT
                ASL     R3                      ;OFFSET OF 20 TIMES UNIT #
MORXMT:         MOVB    7(R2),R4                ;PUT LINE # IN R4
                BIC     #177760,R4              ;PURGE LEFTOVER DATA FROM R4
```

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

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

003656' 017703  176106          RCVRTN: MOV     @RPNTR,R3               ;GET OFFSET
003662' 062767  000002  176100          ADD     #2,RPNTR                ;INCREMENT READ POINTER
003670' 026727  176074  001636'         CMP     RPNTR,#RQ+10            ;TIME TO WRAP?
003676' 001003                          BNE     1$                      ;NO
003700' 012767  001626' 176062          MOV     #RQ,RPNTR               ;YES
003706' 010300                  1$:     MOV     R3,R0                   ;GET OFFSET AGAIN
003710' 006200                          ASR     R0                      ;FORM A ONE WORD
003712' 006200                          ASR     R0                      ;INDEX
003714' 010302                  PASSED: MOV     R3,R2                   ;GET READY
003716' 062702  174064                  ADD     ADDR,R2                 ;TO FORM DEVICE ADDRESS
003722' 032712  020000                  BIT     #020000,(R2)            ;SEE IF BUFFER SILO FULL SET
003726' 001022                          BNE     CONT1                   ;CONTINUE IF OK
003730' 010267  174144                  MOV     R2,CSRA                 ;STORE COMMAND REGISTER ADDRESS
```

```
667  003734' 011267  174142                 MOV     (R2),ACSR               ;STORE CONTENTS OF COMMAND REGISTER
668  003740' 012767  000011  174140          MOV     #11,ERRTYP              ;ILLEGAL INTERRUPT
669                                          ;*************************************************************
670  003746' 104405  000000' 000000          ERROR$,BEGIN,NULL       ;FALSE INTERRUPT
671                                          ;*************************************************************
672  003754' 016767  176036  176036          MOV     TIME,TIMER              ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
673  003762' 052777  010000  174110          BIS     #010000,@CSRA           ;BETTER ENABLE RECEIVER BEFORE LEAVING
674  003770' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
675
676  003774' 010304                  CONT1:  MOV     R3,R4                   ;BEGIN TO FORM
677  003776' 006304                          ASL     R4                      ;A NEW OFFSET
678  004000' 006304                          ASL     R4                      ;INTO EXPECTED RECEIVE DATA TABLE
679  004002' 010405                          MOV     R4,R5                   ;BETTER SAVE IT
680  004004' 006304                          ASL     R4                      ;SO YOU CAN FORM A SILO OFFSET
681  004006' 006304                          ASL     R4
682  004010' 060004                          ADD     R0,R4                   ;OFFSET = 202(OCTAL) TIMES UNIT#
683  004012' 062704  000606'                 ADD     #SILO,R4                ;FORM SILO BASE OFFSET
684  004016' 010401                          MOV     R4,R1                   ;REMEMBER FIRST SILO WORD ADDRESS
685  004020' 016224  000002          NOTDUN: MOV     2(R2),(R4)+             ;MOVE DATA FROM DJ TO SILO BUFFER
686  004024' 005360  001712'                 DEC     RCNT(R0)                ;DECREMENT # OF CHARACTERS RECEIVED
687  004030' 003373                          BGT     NOTDUN                  ;BR IF NOT RECEIVED 65 YET
688  004032' 016760  175746  001712'         MOV     CNTSEL,RCNT(R0)         ;RESET RECEIVE COUNT
689  004040' 062705  000406'                 ADD     #RCVTBL,R5              ;FORM OFFSET INTO EXP RECV DATA TABLE
690  004044' 010567  175722                  MOV     R5,I                    ;SAVE IT IN 'I' AS AN INDEX
691                                                                          ;(RCVTBL BASE ADDRESS FOR THIS UNIT)
692  004050' 116105  000001          NXTCHR: MOVB    1(R1),R5                ;GET LINE NUMBER FROM DATA WORD
693  004054' 042705  177760                  BIC     #177760,R5              ;MASK SO HAVE ONLY LINE #
694  004060' 006305                          ASL     R5                      ;MAKE LINE# EVEN WORD BOUNDARY
695  004062' 066705  175704                  ADD     I,R5                    ;FORM EXPECTED DATA ADDRESS
696  004066' 011567  175734                  MOV     (R5),CHAR               ;-TEMP HLDING FOR BIT SIZING
697  004072' 146767  175726  175726          BICB    MASK,CHAR               ;-CHECK ONLY STRAPPED BITS
698  004100' 026721  175722                  CMP     CHAR,(R1)+              ;-COMPARE EXPECTED VS ACTUAL DATA
699  004104' 001066                          BNE     DATERR                  ;BR IF BAD DATA
700  004106' 105215                  DONE:   INCB    (R5)                    ;INCREMENT EXPECTED DATA
701  004110' 020104                          CMP     R1,R4                   ;HAVE YOU FINISHED ENTIRE BUFFER?
702  004112' 103756                          BLO     NXTCHR                  ;GO BACK IF NOT
703  004114' 005360  001722'                 DEC     ENDCNT(R0)
704  004120' 003011                          BGT     1$
705  004122' 012760  000001  001722'         MOV     #1,ENDCNT(R0)                                           ;MK001
706  004130' 042712  050405                  BIC     #050405,(R2)            ;TURN OFF DJ11
707  004134' 005367  175654                  DEC     ENDFLG
708  004140' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
709  004144' 010205                  1$:     MOV     R2,R5                   ;COPY CSR ADDRESS TO R5
710  004146' 004767  000024                  JSR     PC,TBLSNC               ;SYNCH RCV AND XMT TABLES
711  004152' 052712  050000                  BIS     #50000,(R2)             ;NO- ENABLE SILO FULL INT AND XMTR INT
712  004156' 016062  001742' 000004          MOV     XMITEN(R0),4(R2)        ;TURN XMITTERS BACK ON
713  004164' 016060  001742' 001752'         MOV     XMITEN(R0),XMITSV(R0)   ;SAVE INDICATOR OF LINES ACTIVATED
714  004172' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
715
716                                  ;SYNCHRONIZE RECEIVER AND TRANSMITTER TABLES FOR ONE UNIT
717                                  ;WHOSE CSR ADDRESS IS IN R5
718  004176' 010546                  TBLSNC: MOV     R5,-(SP)                ;CSR ADDRESS
719  004200' 166716  173602                  SUB     ADDR,(SP)
720  004204' 006316                          ASL     (SP)                    ;GET OFFSET (UNIT # TIMES 20 OCTAL)
721  004206' 011646                          MOV     (SP),-(SP)              ;COPY OFFSET
722  004210' 006316                          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$=  104407           2[illegible]#    501     502     547     548
BR1      000012R          231#    489     493
BR2      000013R          232#
BTOD$  = 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$=  104404           281#    755
DECCNT   003536R          616     625     627#
DJLINK   000226R          294#    477
DONE     004106R          700#    768
DROP     004432R          770     [illegible]#
DROPM    004422R          516     770#
DVID1    000014R          233#    402     417     438
ECCMEM=  000100           281#
ENDCNT   001722R          337#    426*    703*    705*
ENDFLG   002014R          373#    476*    535*    549     707*
ENDIT$=  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  0000547R  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:XDJALO,DE2:XDJALO.SEQ/NL:TOC/SOL/CRF/DOC=DDXCOM,XDJALO
RUN-TIME:  1  2  .3 SECONDS
RUN-TIME RATIO:  12/4=2.5
CORE USED:  10K  (20 PAGES)

DOCUMENT PAGES:  20
```