```
                        .REM %

                              IDENTIFICATION
                              --------------

                              PRODUCT CODE: AC-E721H-MC

                              PRODUCT NAME: CXDJAH0 DEC/X11 DJ11 MODULE

                              DATE:         FEB 1979

                              MAINTAINER:   DEC/X11 SUPPORT GROUP

                        THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
                        WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT
                        BY DIGITAL EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT
                        CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT
                        MAY APPEAR IN THIS MANUAL.

                        THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE
                        PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER
                        SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS
                        COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY
                        OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

                        DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR
                        THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
                        NOT SUPPLIED BY DIGITAL.

                      COPYRIGHT (C) 1973,1979 DIGITAL EQUIPMENT CORPORATION
```

```
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: 1154
                        2. OCTAL WORDS: 02202
                        3. OCTAL BYTES: 4405


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
```

```
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
```

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
      CHECK 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.

%

```
                                        IOMOD  <DJAH >,1,1,5,,,0,36
                                        MODULE  140000,DJAH,1,1,5,,,0,36
                                        .TITLE  DJAH DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************************
000000-                         BEGIN:
000000- 045104  044101      040 MODNAM: .ASCII  /DJAH / ;MODULE NAME.
000005-     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006- 000001                  ADDR1:  1+0                     ;1ST DEVICE ADDR.
000010- 000001                  VECTOR: 1+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- 002054-                 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-                         SRADR:                          ;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- 002410-                 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:
                                ;*****************************************************************************

                                ;SOME VARIABLES AND CONSTANTS UNIQUE TO THIS ROUTINE
000224- 004567  003276          DJLINK: JSR     R5,RCVINT               ;UNIT 0 RECEIVER
000230- 000000                          0
000232- 004567  002734                  JSR     R5,XMTINT               ;UNIT 0 TRANSMITTER
000236- 000000                          0
000240- 004567  003262                  JSR     R5,RCVINT               ;UNIT 1 RECEIVER
000244- 000010                          10
000246- 004567  002720                  JSR     R5,XMTINT               ;UNIT 1 TRANSMITTER
000252- 000010                          10
000254- 004567  003246                  JSR     R5,RCVINT               ;UNIT 2 RECEIVER
000260- 000020                          20
000262- 004567  002704                  JSR     R5,XMTINT               ;UNIT 2 TRANSMITTER
000266- 000020                          20
000270- 004567  003232                  JSR     R5,RCVINT               ;UNIT 3 RECEIVER
000274- 000030                          30
000276- 004567  002670                  JSR     R5,XMTINT               ;UNIT 3 TRANSMITTER
000302- 000030                          30

                        000000          OPEN=0

000304- 000040          XMTTBL:.BLKW 32.                ;A TABLE FOR RECORDING CHARACTERS WHICH HAVE
                                                        ;BEEN TRANSMITTED
000404- 000100          RCVTBL:.BLKW 64.                ;A TABLE FOR RECORDING CHARACTERS WHICH YOU
                                                        ;EXPECT TO RECEIVE
000604- 000404          SILO:   .BLKW 260.              ;A BUFFER AREA TO STORE THE DATA RECORDED IN
                                                        ;THE DJ'S RECEIVER SILO
001614- 000004          XQ:     .BLKW 4.                ;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
                                                        ;TRANSMIT INTERRUPT IS IN "PIRQ"
001624- 000004          RQ:     .BLKW 4.                ;A QUEUE FOR SAVING THE DEVICE OFFSET WHILE A
                                                        ;RECEIVER INTERRUPT IS IN "PIRQ"
001634- 000020          ERRQ:   .BLKW   16.
001674- 000000          ERRQI:  OPEN
001676- 000000          ERRQO:  OPEN

001700- 000101  000101  000101  XCNT:.WORD 101,101,101,101  ;NUMBER OF CHARACTERS LEFT TO BE TRANSMITTED DURING
001706- 000101                                              ;THIS INTERRUPT. A WORD FOR EACH DEVICE

001710- 000101  000101  000101  RCNT:.WORD 101,101,101,101  ;NUMBER OF CHARACTERS TO BE PROCESSED BY THE
001716- 000101                                              ;RECEIVE ROUTINE. A WORD FOR EACH DEVICE

001720- 000200  000200  000200  ENDCNT:.WORD 200,200,200,200        ;NUMBER OF "65 CHARACTER CHUNKS"
001726- 000200                                                      ;TO BE PROCESSED BY THE MODULE PER PASS

001730- 177777  177777  177777  XMTSEL: .WORD   177777,177777,177777,177777     ;LINES TO BE RUN
```

```
001736' 177777
;
001740' 177777  177777  177777  XMITEN:.WORD 177777,177777,177777,177777        ;WORD USED TO FILL TRANSMITTER COMMAND
001746' 177777
                                                                               ;REGISTER 16XXX2,CORRESPONDS TO
                                                                               ;A ONE FOR EACH ACTIVE LINE
;
001750' 000000  000000  000000  XMITSV: .WORD 0,0,0,0                    ;SAVE LINES ACTIVATED IN CASE OF HANG
001756' 000000
;
;
;
;
;
001760' 001614'                 XPNTW:  XQ                              ;POINTERS USED FOR THE BOOKKEEPING OF
001762' 001614'                 XPNTR:  XQ                              ;THE TRANSMIT QUEUE
001764' 001624'                 RPNTW:  RQ                              ;POINTERS USED FOR THE BOOKKEEPING OF
001766' 001624'                 RPNTR:  RQ                              ;THE RECEIVE QUEUE
;
001770' 000000                  I:      000                             ;TEMPORARY STORAGE LOCATION,USED DURING RECEIVE DATA CHE
001772' 000001                  SELMSK: 1                               ;MASK TO CHECK FOR UNIT SELECTED
001774' 000000                  SELECT: 0
001776' 100000                  EXPLIN: 100000                          ;USED DURING INITIALIZATION,THE ONE REPRESENTS THE
                                                                        ;DONE BIT IN THE EXPECTED RECEIVE DATA TABLE
002000' 000200                  ENDSEL: 200                             ;REFILL OF 'ENDSEL' WORD AFTER ENDPASS
002002' 000101                  CNTSEL: 101                             ;REFILL OF 'RCNT' AND 'XCNT' AFTER AN INTERRUPT SERVICE
                                                                        ;NOT VARIABLE SINCE 65. CHARS REQD TO GET SILO FULL INT
002004' 000000                  VCT:    OPEN                            ;HOLDS VECTOR ADDRESS DURING SETUP ERROR PRINTOUT
002006' 000000                  LINK:   OPEN                            ;HOLDS LINK ADDRESS DURING SETUP ERROR PRINTOUT
002010' 000000                  SCSR:   OPEN
002012' 000000                  ENDFLG: OPEN
002014' 000110                  TIME:   110                             ;NUMBER OF DJ11'S STILL RUNNING
002016' 000000                  TIMER:  0                               ;WATCHDOG TIMER OUTER LOOP VALUE
002020' 000000                  TIME1:  0
002022' 000000                  MASK:   0                                       ;-DATA BIT LENGTH (5-8) MASK
002024' 000000                  CHAR:   0                                       ;-TEMP HOLDING FOR MASK & ITERATION COUNT &
                                                                                ;-DATA COMPARSION DURRING TEST.
002026' 017000                  TMETBL: 17000                                   ;-9600  BPS VARIABLE
002030' 011700                          11700                                   ;-4800  BPS ITERATION
002032' 006000                          6000                                    ;-2400  BPS COUNTS
002034' 005100                          5100                                    ;-1800  BPS PER
002036' 003000                          3000                                    ;-1200  BPS LINE
002040' 001600                          1600                                    ;-600   BPS SPEEDS
002042' 000750                          750                                     ;-300   BPS
002044' 000350                          350                                     ;-150   BPS
002046' 000300                          300                                     ;-134.5 BPS
002050' 000260                          260                                     ;-110   BPS
002052' 000165                          165                                     ;-75    BPS
;
;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
002054' 012767  000101  176036  START:  MOV     #65.,INTR               ;65 INTERRUPTS/ITERATION
002062' 012767  000100  176024          MOV     #64.,WDTO               ;64 WORDS TO MEM/ITERATION
002070' 012767  000100  176020          MOV     #64.,WDFR               ;64 WORDS FROM MEM/ITERATION
```



```
002076' 016767  175712  177670  MOV     DVID1,SELECT
002104' 012704  000304'         MOV     #XMTTBL,R4
002110' 005024                  1$:     CLR     (R4)+                   ;CLEAR XMT TABLE
002112' 020427  000404'                 CMP     R4,#RCVTBL
002116' 103774                          BLO     1$
002120' 016702  175672                  MOV     SR1,R2                  ;-SET UP TO FIND ITERATION CNT
002124' 042702  000037                  BIC     #37,R2                  ;-STRIP UNWANTED BITS
002130' 012703  002026'                 MOV     #TMETBL,R3              ;-POINT TO ITERATION TABLE
002134' 006302                  2$:     ASL     R2                      ;-SHIFT TO FIND COUNT
002136' 005702                          TST     R2                      ;-IS THIS IT?
002140' 001402                          BEQ     3$                      ;-YES LEAVE NOW
002142' 005723                          TST     (R3)+                   ;-POINT TO NXT IN TBL
002144' 000773                          BR      2$                      ;-TRY AGAIN
002146' 011367  177626          3$:     MOV     (R3),ENDSEL             ;SET UP COUNT
002152' 012700  000004                  MOV     #4,R0
002156' 005001                          CLR     R1
002160' 016761  177616  001710' 4$:     MOV     CNTSEL,RCNT(R1)
002166' 016761  177610  001700'         MOV     CNTSEL,XCNT(R1)
002174' 016761  177600  001720'         MOV     ENDSEL,ENDCNT(R1)
002202' 062701  000002                  ADD     #2,R1
002206' 005300                          DEC     R0
002210' 001363                          BNE     4$
002212' 005767  175620                  TST     ICONT                   ;IS THIS THE FIRST TIME THRU ?
002216' 001031                          BNE     9$                      ;NO BRANCH
002220' 016767  177554  175610          MOV     ENDSEL,ICONT            ;YES SET THE COUNT TO ICONT
002226' 005267  175604                  INC     ICONT                   ;MUST BE GREATER BY ONE BECAUSE
                                                                        ;LAST ENDIT XALL IS DONE AFTER
                                                                        ;A TEST TO SEE IF ALL DEVS
                                                                        ;FUNCTIONED BEFORE WATCHDOG TIMER
                                                                        ;TIMED OUT
002232' 016701  175556                  MOV     DVID1,R1                ;SAVE DVC IN R1
002236' 001324                          BNE     1$                      ;IF SET UP OK - BRANCH
002240' 104410  000000'                 ENDS,BEGIN              ;
002244' 006201                  11$:    ASR     R1                      ;ANY DEVS LEFT ??
002246' 001720                          BEQ     1$                      ;NO - GOODBYE
002250' 066767  177524  175560          ADD     ENDSEL,ICONT            ;YUP - SET ADDITIONAL COUNT
002256' 062767  000101  175634          ADD     #65.,INTR               ;65 MORE TRUPS
002264' 062767  000100  175622          ADD     #64.,WDTO               ;64. MORE WORDS TO
002272' 062767  000100  175616          ADD     #64.,WDFR               ;64. MORE WORDS FROM
002300' 000761                          BR      11$                     ;GO CHECK FOR MORE
002302' 012767  100000  177466  9$:     MOV     #100000,EXPLIN          ;SETUP TO PRESET THE RECEIVE TABLE
002310' 012704  000404'                 MOV     #RCVTBL,R4              ;GET START ADDRESS OF RECEIVE TABLE
002314' 016724  177456          5$:     MOV     EXPLIN,(R4)+            ;LOAD RECEIVE TABLE WITH A DONE BIT AND LINE #
002320' 105267  177453                  INCB    EXPLIN+1                ;INCREMENT TO THE NEXT LINE NUMBER
002324' 042767  070000  177444          BIC     #070000,EXPLIN          ;MAKE SURE LINE # NEVER GREATER THAN FIFTEEN
002332' 020427  000604'                 CMP     R4,#SILO                ;SEE IF DONE PRESETTING
002336' 103766                          BLO     5$                      ;BRANCH BACK IF NOT
002340' 012767  177400  177456          MOV     #177400,CHAR            ;-PRESET MASK FOR 8 BIT DATA
002346' 005767  175444                  TST     SR1                     ;-8 BIT ?
002352' 001413                          BEQ     7$                      ;-IF YES- SET IT UP
002354' 016767  175436  177440          MOV     SR1,MASK                ;-GET OPERATOR INPUT
002362' 042767  177770  177432          BIC     #177770,MASK            ;-ELIMINATE ITERATION BAUD COUNT
002370' 006267  177430          6$:     ASR     CHAR                    ;-SHIFT TO RIGHT-
002374' 006267  177422                  ASR     MASK                    ;-SIZE MASK.
002400' 001373                          BNE     6$                      ;-DONE?
002402' 116767  177416  177412  7$:     MOVB    CHAR,MASK               ;-SAVE IT.
```

```
 432  002410' 005767  177360           RESTRT: TST     SELECT                  ;ANY DJ'S SELECTED?
 433  002414' 001404                           BEQ     1$                      ;BR IF NO
 434  002416' 032767  177760  177350           BIT     #177760, SELECT         ;UNITS OTHER THAN 0-3 SELECTED?
 435  002424' 001402                           BEQ     2$                      ;NO- OK
 436  002426'                          1$:
 437  002426' 104410  000000'                 ENDS,BEGIN              ;UNIT SELECTION (DVIDI) INCORRECT
 438  002432' 012767  001614' 177320   2$:     MOV     #XQ,XPNTW               ;SETUP QUEUE POINTERS
 439  002440' 012767  001614' 177314           MOV     #XQ,XPNTR
 440  002446' 012767  001624' 177310           MOV     #RQ,RPNTW
 441  002454' 012767  001624' 177304           MOV     #RQ,RPNTR
 442  002462' 012767  001634' 177204           MOV     #ERRQ,ERRQI
 443  002470' 012767  001634' 177200           MOV     #ERRQ,ERRQO
 444  002476' 005067  177310                   CLR     ENDFLG
 445  002502' 012701  000224'                 MOV     #DJLINK,R1              ;GET ADDRESS OF DJ'S LINKING TABLE
 446  002506' 016700  175276                   MOV     VECTOR,R0               ;GET VECTOR ADDRESS
 447  002512' 016705  175270                   MOV     ADDR,R5                 ;GET CSR ADDRESS
 448  002516' 012767  000001  177246           MOV     #1,SELMSK
 449  002524' 036767  177242  177242   NXTDEV: BIT     SELMSK,SELECT           ;HAS THIS DEVICE BEEN SELECTED?
 450  002532' 001530                           BEQ     1$                      ;NO- BRANCH
 451  002534' 010120                           MOV     R1,(R0)+                ;YES- LOAD RCV INT VECTOR
 452  002536' 116720  175250                   MOVB    BR4,(R0)+               ;LOAD RCV BR
 453  002542' 105020                           CLRB    (R0)+
 454  002544' 062701  000006                   ADD     #6,R1
 455  002550' 010120                           MOV     R1,(R0)+                ;LOAD XMT INT VECTOR
 456  002552' 116720  175234                   MOVB    BR4,(R0)+               ;LOAD XMT BR
 457  002556' 105020                           CLRB    (R0)+
 458  002560' 062701  000006                   ADD     #6,R1
 459  002564' 004767  001334                   JSR     PC,TBLSNC               ;SYNCH RCV AND XMT TABLES
 460  002570' 012715  000014                   MOV     #14,(R5)                ;CLEAR DJ BUFFER AND UARTS
 461  002574' 012704  000010                   MOV     #10,R4                  ;SETUP WAIT LOOP COUNTER
 462  002600' 032715  000020           10$:    BIT     #20,(R5)                ;"BUSY CLEAR" GONE LOW YET?
 463  002604' 001456                           BEQ     11$                     ;YES- BRANCH
 464  002606' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 465  002612' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 466  002616' 005304                           DEC     R4                      ;WAIT LOOP COUNTER
 467  002620' 001367                           BNE     10$                     ;LOOP IF NOT TIMED OUT
 468  002622' 010167  177160                   MOV     R1,LINK                 ;SAVE INFO DURING ERROR (CAN BE DONE
 469  002626' 010067  177152                   MOV     R0,VCT                  ;SINCE THIS IS A SEQUENTIAL LOOP
 470  002632' 010567  177152                   MOV     R5,SCSR                 ;AND INFO NOT USED ELSEWHERE)
 471  002636' 036767  177130  177130           BIC     SELMSK,SELECT           ;CLEAR UNIT SELECTED BIT - DROP UNIT
 472  002644' 012567  175230                   MOV     R5,CSRA                 ;SETUP FOR ERROR PRINTOUT
 473  002650' 011567  175226                   MOV     @R5,ACSR
 474  002654' 005015                           CLR     @R5                     ;DISABLE UNIT
 475  002656' 012767  000016  175222           MOV     #16,ERRTYP              ;BUSY STUCK
 476                                   ;*****************************************************************
 477  002664' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;BUSY CLEAR DIDN'T CLEAR IN TIME
 478                                   ;*****************************************************************
 479  002672' 104403  000000' 004350'         MSGN$,BEGIN,DROPM       ;ASCII MESSAGE CALL WITH COMMON HEADER
 480                                                                   ;"UNIT DROPPED"
 481  002700' 016767  177110  177110           MOV     TIME,TIMER              ;RESET TIMER DUE TO PRINTOUT DELAY
 482  002706' 005767  177062                   TST     SELECT                  ;ANY DJ'S STILL SELECTED?
 483  002712' 001002                           BNE     12$                     ;YES- BRANCH
 484  002714' 104410  000000'                 END$,BEGIN              ;ALL UNITS DROPPED
 485  002720' 016701  177062           12$:    MOV     LINK,R1
 486  002724' 016700  177054                   MOV     VCT,R0
 487  002730' 016705  177054                   MOV     SCSR,R5
```



```
 488  002734' 062705  000010                   ADD     #10,R5
 489  002740' 000433                           BR      2$
 490  002742' 012715  050405           11$:    MOV     #50405,(R5)             ;TURN ON RCVR AND XMIT ENABLE
 491  002746' 010503                           MOV     R5,R3
 492  002750' 166703  175032                   SUB     ADDR,R3
 493  002754' 006203                           ASR     R3
 494  002756' 006203                           ASR     R3                      ;GET OFFSET (UNIT # TIMES 2)
 495  002760' 016363  001730' 001740'          MOV     XMTSEL(R3),XMITEN(R3)
 496  002766' 016363  001740' 001750'          MOV     XMITEN(R3),XMITSV(R3)   ;SAVE IN CASE OF HANG TO SHOW ACTIVE LINES
 497  002774' 016365  001740' 000004           MOV     XMITEN(R3),4(R5)        ;TURN ON XMTR
 498  003002' 005267  177004                   INC     ENDFLG                  ;COUNT UNITS RUNNING
 499  003006' 062705  000010                   ADD     #10,R5
 500  003012' 000406                           BR      2$
 501  003014' 062701  000014           1$:     ADD     #14,R1
 502  003020' 062700  000010                   ADD     #10,R0
 503  003024' 062705  000010                   ADD     #10,R5
 504  003030' 006367  176736           2$:     ASL     SELMSK                  ;NEXT UNIT
 505  003034' 026727  176732  000020           CMP     SELMSK,#20              ;DONE?
 506  003042' 002630                           BLT     NXTDEV                  ;NO- LOOP
 507  003044' 016767  176744  176744           MOV     TIME,TIMER
 508  003052' 005067  176742                   CLR     TIME1                   ;YES- DELAY
 509  003056'                          3$:
 510  003056' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 511  003062' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 512  003066' 005767  176720                   TST     ENDFLG                  ;DJ11'S DONE?
 513  003072' 001002                           BNE     4$                      ;NO- BRANCH
 514  003074' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
 515                                                                   ;MONITOR SHALL TEST END OF PASS
 516  003100' 005367  176714           4$:     DEC     TIME1                   ;COUNT TIMER
 517  003104' 001364                           BNE     3$                      ;TO CATCH
 518  003106' 005367  176704                   DEC     TIMER                   ;HUNG DJ11'S
 519  003112' 001361                           BNE     3$
 520  003114' 104403  000000' 004354'         MSGN$,BEGIN,HUNGM       ;ASCII MESSAGE CALL WITH COMMON HEADER
 521  003122' 016705  174660                   MOV     ADDR,R5
 522  003126' 012767  000001  176636           MOV     #1,SELMSK
 523  003134' 036767  176632  176632   5$:     BIT     SELMSK,SELECT
 524  003142' 001401                           BEQ     6$                      ;BRANCH IF NOT SELECTED
 525  003144' 005015                           CLR     @R5                     ;IF SELECTED CLEAR THE CSR
 526  003146' 062705  000010           6$:     ADD     #10,R5                  ;NEXT CSR ADDRESS
 527  003152' 006367  176614                   ASL     SELMSK                  ;NEXT UNIT SELECT BIT
 528  003156' 026727  176610  000020           CMP     SELMSK,#20              ;ALL UNITS CHECKED?
 529  003164' 002763                           BLT     5$                      ;BRANCH IF NOT
 530  003166' 104410  000000'                 END$,BEGIN              ;DJ11 HUNG
 531
 532                                   ;ROUTINE TO HANDLE TRANSMITTER INTERRUPTS
 533  003172' 010246                   XMTINT: MOV     R2,-(SP)                ;SAVE REG 2 ON STACK
 534  003174' 011502                           MOV     (R5),R2                 ;GET OFFSET INTO R2
 535  003176' 066702  174604                   ADD     ADDR,R2                 ;FORM DEVICE ADDRESS
 536  003202' 042712  040000                   BIC     #040000,(R2)            ;TURN OFF XMT INT ENABLE WHILE YOU SERVICE
 537  003206' 011577  176546                   MOV     (R5),@XPNTW             ;SAVE OFFSET IN QUEUE
 538  003212' 062767  000002  176540           ADD     #2,XPNTW                ;INCREMENT IN POINTER
 539  003220' 026727  176534  001624'          CMP     XPNTW,#XQ+10            ;TIME TO WRAP?
 540  003226' 001003                           BNE     1$                      ;NO
 541  003230' 012767  001614' 176522           MOV     #XQ,XPNTW               ;YES
 542  003236' 012602                   1$:     MOV     (SP)+,R2                ;RESTORE R2
 543  003240' 012605                           MOV     (SP)+,R5                ;RESTORE R5
```

```
                        ;-----------------------------------------------------------------
003242' 000004  000000' 003250'         PIRQ$,BEGIN,XMTRTN               ; QUEUE UP TO CONTINUE AT XMTRTN AND RTI
                        ;-----------------------------------------------------------------

003250' 017703  176506          XMTRTN: MOV     @XPNTR,R3               ;GET OFFSET
003254' 062767  000002  176500          ADD     #2,XPNTR                ;INCREMENT OUT POINTER
003262' 026727  176474  001624'         CMP     XPNTR,#XQ+10            ;TIME TO WRAP?
003270' 001003                          BNE     1$                      ;NO
003272' 012767  001614' 176462          MOV     #XQ,XPNTR               ;YES
003300' 010302                  1$:     MOV     R3,R2                   ;GET READY- COPY OFFSET
003302' 066702  174500                  ADD     ADDR,R2                 ;TO FORM DEVICE ADDRESS
003306' 005712                          TST     (R2)                    ;XMT READY BIT SET?
003310' 100424                          BMI     XMTOK                   ;BR IF OK
003312' 013724  174562                  MOV     @2,CSRA                 ;STORE COMMAND REGISTER ADDRESS
003316' 011567  174560                  MOV     (R2),ACSR               ;STORE CONTENTS OF COMMAND REGISTER
003322' 012767  000011  174556          MOV     #11,ERRTYP              ;ILLEGAL INTERRUPT
                        ;*****************************************************************
003330' 104405  000000' 000000          ERDER$,BEGIN,NULL       ;XMIT READY NOT SET ON XMIT INTERRUPT
                        ;*****************************************************************
003336' 016767  176452  176452          MOV     TIME,TIMER              ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
003344' 052777  040000  174526          BIS     #040000,@CSRA           ;BETTER REENABLE TRANSMITTERS
003352' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

003356' 010300                  XMTOK:  MOV     R3,R0                   ;PREPARE A ONE WORD OFFSET
003360' 006200                          ASR     R0                      ;BEGIN THE OFFSET WITH ONE SHIFT
003362' 006200                          ASR     R0                      ;COMPLETE THE OFFSET WITH 2ND SHIFT
003364' 006303                          ASL     R3                      ;OFFSET OF 20 TIMES UNIT #
003366' 112204  000007          MORXMT: MOVB    7(R2),R4                ;PUT LINE # IN R4
003372' 042704  177760                  BIC     #177760,R4              ;PURGE LEFTOVER DATA FROM R4
003376' 060304                          ADD     R3,R4                   ;FORM UNIQUE LINE/CHAR OFFSET
003400' 116467  000304' 176416          MOVB    XMTTBL(R4),CHAR         ;-TEMP HLD CHAR
003406' 146767  176410  176410          BICB    MASK,CHAR               ;-FOR MASKING
003414' 116762  176404  000006          MOVB    CHAR,6(R2)              ;-TRANSMIT CHARACTER
003422' 105264  000304'                 INCB    XMTTBL(R4)              ;ADVANCE TO NEXT CHARACTER
003426' 001016                          BNE     DECCNT                  ;AVOID NEXT SECTION OF CODE
                                                                        ;IF SENT CHARS NOT WRAPPED TO ZERO
003430' 012701  000001                  MOV     #1,R1                   ;BIT POSITION OF THE ONE IS SAME AS XMT LINE #
003434' 042704  177760                  BIC     #177760,R4              ;LEAVE ONLY LINE # IN R4
003440' 001403                          BEQ     LINOFF                  ;LINE 0 IS DONE,TURN IT OFF
003442' 006301                  1$:     ASL     R1                      ;SHIFT XMT BIT TO NEXT XMTTER LINE POSITION
003444' 005304                          DEC     R4                      ;DECREMENT ACTUAL LINE NUMBER
003446' 001375                          BNE     1$                      ;BRANCH BACK IF NOT DONE
003450' 040160  001740'         LINOFF: BIC     R1,XMITEN(R0)           ;TURN OFF THE INDIVIDUAL  XMTTER LINE
003454' 001003                          BNE     DECCNT                  ;ARE ALL LINES DONE NOW?
003456' 010160  001730' 001740'         MOV     XMTSEL(R0),XMITEN(R0)   ;IF YES, RESET ENABLE WORD
003464' 005360  001700'         DECCNT: DEC     XCNT(R0)                ;DECREMENT CHARACTER COUNT
003470' 003403                          BLE     OFF                     ;HAVE YOU SENT 65? BR IF YES
003472' 005712                          TST     (R2)                    ;ADDITIONAL XMTTRS READY
003474' 100734                          BMI     MORXMT                  ;BR IF YES
003476' 000407                          BR      XMTOUT                  ;GO OUT AND WAIT FOR ANOTHER LINE TO INT
003500' 005062  000004          OFF:    CLR     4(R2)                   ;TURN OFF XMTTER
003504' 016760  176272  001700'         MOV     CNTSEL,XCNT(R0)         ;RESET CHAR COUNT
003512' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003516' 052712  040000          XMTOUT: BIS     #040000,(R2)            ;ENABLE MORE XMITR INTERRUPTS
003522' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
                        ;ROUTINE TO HANDLE RECEIVER INTERRUPTS
003526' 010246          RCVINT: MOV     R2,-(SP)                ;SAVE R2 ON STACK
003530' 011502                  MOV     (R5),R2                 ;GET OFFSET
003532' 066702  174250          ADD     ADDR,R2                 ;FORM DEVICE ADDRESS
003536' 042712  010000          BIC     #010000,(R2)            ;TURN OFF INTERRUPTS
003542' 011577  176216          MOV     (R5),@RPNTW             ;SAVE OFFSET IN QUEUE
003546' 062767  000002  176210          ADD     #2,RPNTW                ;INCREMENT WRITE POINTER
003554' 026727  176204  001634'         CMP     RPNTW,#RQ+10            ;TIME TO WRAP?
003562' 001003                          BNE     1$                      ;NO
003564' 012767  001624' 176172          MOV     #RQ,RPNTW               ;YES
003572' 012602                  1$:     MOV     (SP)+,R2                ;RESTORE R2
003574' 012605                          MOV     (SP)+,R5                ;RESTORE R5
                        ;-----------------------------------------------------------------
003576' 000004  000000' 003604'         PIRQ$,BEGIN,RCVRTN              ; QUEUE UP TO CONTINUE AT RCVRTN AND RTI
                        ;-----------------------------------------------------------------

003604' 017703  176156          RCVRTN: MOV     @RPNTR,R3               ;GET OFFSET
003610' 062767  000002  176150          ADD     #2,RPNTR                ;INCREMENT READ POINTER
003616' 026727  176144  001634'         CMP     RPNTR,#RQ+10            ;TIME TO WRAP?
003624' 001003                          BNE     1$                      ;NO
003626' 012767  001624' 176132          MOV     #RQ,RPNTR               ;YES
003634' 010300                  1$:     MOV     R3,R0                   ;GET OFFSET AGAIN
003636' 006200                          ASR     R0                      ;FORM A ONE WORD
003640' 006200                          ASR     R0                      ;INDEX
003642' 010302                  PASSED: MOV     R3,R2                   ;GET READY
003644' 066702  174136                  ADD     ADDR,R2                 ;TO FORM DEVICE ADDRESS
003650' 032712  020000                  BIT     #020000,(R2)            ;SEE IF BUFFER SILO FULL SET
003654' 001027                          BNE     CONT1                   ;CONTINUE IF OK
003656' 013724  174216                  MOV     @2,CSRA                 ;STORE COMMAND REGISTER ADDRESS
003662' 011567  174214                  MOV     (R2),ACSR               ;STORE CONTENTS OF COMMAND REGISTER
003666' 012767  000011  174212          MOV     #11,ERRTYP              ;ILLEGAL INTERRUPT
                        ;*****************************************************************
003674' 104405  000000' 000000          ERDER$,BEGIN,NULL       ;FALSE INTERRUPT
                        ;*****************************************************************
003702' 016767  176106  176106          MOV     TIME,TIMER              ;RESET TIMER DUE TO MESSAGE PRINTOUT DELAY
003710' 052777  010000  174162          BIS     #010000,@CSRA           ;BETTER ENABLE RECEIVER BEFORE LEAVING
003716' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

003722' 010304                  CONT1:  MOV     R3,R4                   ;BEGIN TO FORM
003724' 006304                          ASL     R4                      ;A NEW OFFSET
003726' 006304                          ASL     R4                      ;INTO EXPECTED RECEIVE DATA TABLE
003730' 010405                          MOV     R4,R5                   ;BETTER SAVE IT
003732' 006304                          ASL     R4                      ;SO YOU CAN FORM A SILO OFFSET
003734' 006304                          ASL     R4
003736' 060004                          ADD     R0,R4                   ;OFFSET = 202(OCTAL) TIMES UNIT#
003740' 062704  000604'                 ADD     #SILO,R4                ;FORM SILO BASE OFFSET
003744' 010401                          MOV     R4,R1                   ;REMEMBER FIRST SILO WORD ADDRESS
003746' 016224  000002          NOTDUN: MOV     2(R2),(R4)+             ;MOVE DATA FROM DJ TO SILO BUFFER
003752' 005360  001710'                 DEC     RCNT(R0)                ;DECREMENT # OF CHARACTERS RECEIVED
003756' 003373                          BGT     NOTDUN                  ;BR IF NOT RECEIVED 65 YET
003760' 016760  176016  001710'         MOV     CNTSEL,RCNT(R0)         ;RESET RECEIVE COUNT
003766' 062705  000404'                 ADD     #RCVTBL,R5              ;FORM OFFSET INTO EXP RECV DATA TABLE
003772' 010567  175772                  MOV     R5,I                    ;SAVE IT IN 'I' AS AN INDEX
                                                                        ;(RCVTBL BASE ADDRESS FOR THIS UNIT)
003776' 116105  000001          NXTCHR: MOVB    1(R1),R5                ;GET LINE NUMBER FROM DATA WORD
004002' 042705  177760                  BIC     #177760,R5              ;MASK SO HAVE ONLY LINE #
```

```
                                        ASL     R5                      ;MAKE LINE# EVEN WORD BOUNDARY
                                        ADD     I6R5                    ;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     ENDSEL,ENDCNT(R0)
                                        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)
                                        ADD     #RCVTBL,(SP)
                                        ADD     #XMITBL,2(SP)
                                        MOV     #16.,-(SP)              ;COUNTER
                                1$:     MOVB    @4(SP),@2(SP)           ;MOVE XMIT TABLE CONTENTS INTO DATA
                                                                        ;PORTION OF RCV TABLE
                                        INC     4(SP)
                                        ADD     #2,2(SP)
                                        DEC     (SP)
                                        BNE     1$
                                        CMP     (SP)+,(SP)+             ;FIX SP
                                        TST     (SP)+
                                        RTS     PC

                                ;ERROR HANDLER FOR DATA ERRORS
                                DATERR: MOV     R2,CSRA                 ;STORE COMMAND REGISTER ADDRESS
                                        MOV     R5,BADR                 ;STORE ADDRESS OF TEST DATA
                                        TST     -(R1)                   ;RESET R1 BACK TO THE ADDRESS OF THE FAIL DATA
                                        MOV     R1,WASADR               ;STORE ADDRESS OF XMITED DATA
                                        MOV     (R5),ASB                ;STORE EXPECTED RECEIVE DATA
                                        MOV     (R1)+,AWAS              ;STORE ACTUAL RECEIVED DATA AND RESET R1 TO RIGH
                                        MOV     ERRQI,R2
                                        BICB    MASK,ASB                ;-MASK OUT STRAPPED OFF BITS
                                        MOV     R0,(R2)+
                                        MOV     R1,(R2)+
                                        MOV     R4,(R2)+
                                        MOV     R5,(R2)+
```



```
                                        CMP     R2,#ERRQ+32.
                                        BLO     1$
                                        MOV     #ERRQ,R2
                                1$:     MOV     R2,ERRQI
                                        ;*************************************************************
                                        DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*************************************************************
                                        MOV     TIME,TIMER              ;RESET TIMER DUE TO PRINTOUT DELAY
                                        MOV     ERRQO,R2
                                        MOV     (R2)+,R0
                                        MOV     (R2)+,R1
                                        MOV     (R2)+,R4
                                        MOV     (R2)+,R5
                                        CMP     R2,#ERRQ+32.
                                        BLO     2$
                                        MOV     #ERRQ,R2
                                2$:     MOV     R2,ERRQO
                                        MOV     CSRA,R2
                                        BR      DONE                    ;NOW CONTINUE CHECKING DATA

                                DROPM:  DROP
                                        -1
                                HUNGM:  HUNG
                                        -1

                                DROP:   .ASCIZ  /UNIT DROPPED/

                                HUNG:   .ASCIZ  /UNIT HUNG/

                                .END
```

```
ACSR    000102R           252#    473*    558*    629*
ADDR    000006R           218#    447     492     521     535     554     603     625     681
ADDR22= 001000            270#
ASB     000106R           256#    704*    707*
ASTAT   000104R           254#
AWAS    000110R           257#    705*
BEGIN   000000R           215#    408     437     464     465     477     479     484     510     511     514     520     530
                          545     561     565     596     598     613     632     636     670     676     717
BIT0    = 000001          270#
BIT1    = 000002          270#
BIT10   = 002000          270#
BIT11   = 004000          270#
BIT12   = 010000          270#
BIT13   = 020000          270#
BIT14   = 040000          270#
BIT15   = 100000          270#
BIT2    = 000004          270#
BIT3    = 000010          270#
BIT4    = 000020          270#
BIT5    = 000040          270#
BIT6    = 000100          270#
BIT7    = 000200          270#
BIT8    = 000400          270#
BIT9    = 001000          270#
BREAK$= 104407            270#    464     465     510     511
BR1     000012R           220#    452     456
BR2     000013R           221#
BTOD$ = 104421            270#
CDATA$= 104412            270#
CHAR    002024R           356#    423*    428*    431     574*    575*    576     658*    659*    660
CNTSEL  002002R           346#    392     393     595     650
CONFIG  000056R           240#
CONT1   003722R           627     638#
CSRA    000100R           250#    472*    557*    564*    628*    635*    700*    729
DATCK$= 104411            270#
DATERR  004210R           661     700#
DATER$= 104404            270#    717
DECCNT  003464R           578     587     589#
DJLINK  000224R           224#    445
DONE    004034R           662#    730
DROP    004360R           732     737#
DROPM   004350R           479     732#
DVID1   000014R           222#    376     406
ENDCNT  001720R           315#    394*    665*    667*
ENDFLG  002012R           351#    444*    498*    512     669*
ENDIT$= 104413            270#    514
ENDSEL  002000R           345#    380*    394     400     411     667
ENDS  = 104410            270#    408     437     484     530
ERRQ    001634R           301#    442     443     712     714     725     727
ERRQI   001674R           302#    442*    706     715*
ERRQO   001676R           303#    443*    720     728*
ERRTYP  000106R           255#    475*    559*    630*
EXIT$ = 104400            270#    565     596     598     636     670     676
EXPLIN  001776R           343#    416*    418     419*    420*
GETPA$= 104415            270#
GWBUP$= 104414            270#
```



```
HRDCNT  000044R           235#
HRDER$= 104405            270#    477     561     632
HRDPAS  000050R           237#
HUNG    004375R           734     740#
HUNGM   004354R           520     734#
I       001770R           340#    652*    657
ICONT   000036R           232#    398     400*    401*    411*
ICOUNT  000040R           233#
IDNUM   000122R           262#
INIT    000030R           229#
INTR    000120R           261#    373*    412*
LINK    002006R           349#    468*    485
LINOFF  003450R           582     586#
MAP22$= 104416            270#
MASK    002022R           355#    426*    427*    429*    431*    575     659     707
MODNAM  000000R           216#
MODSP   000224R           230     268#
MORXMT  003366R           571#    592
MSGN$ = 104403            270#    479     520
MSGS$ = 104402            270#
MSG$  = 104401            270#
NOTDUN  003746R           647#    649
NULL  = 000000            270#    477     561     632
NXTCHR  003776R           654#    664
NXTDEV  002524R           449#    506
OFF     003500R           590     594#
OPEN  = 000000            217     223     224     225     226     243     244     245     246     247     248     249     250
                          252     254     256     257     259     260     261     270#    289#    302     303     348     349
                          350     351
OTOA$ = 104420            270#
PASCNT  000034R           231#
PASSED  003642R           624#
PIRQ$ = 000004            270#    545     613
POPSP = 005726            270#
POPSP2= 022626            270#
PRTY  = 000000            221     270#
PRTY0 = 000000            270#
PRTY1 = 000040            270#
PRTY2 = 000100            270#
PRTY3 = 000140            270#
PRTY4 = 000200            270#
PRTY5 = 000240            220     270#
PRTY6 = 000300            270#
PRTY7 = 000340            270#
PS    = 177776            270#
PSW   = 177776            270#
PUSH  = 005746            270#
PUSH2 = 024646            270#
RAND$ = 104417            270#
RANNUM  000054R           239#
RCNT    001710R           310#    392*    648*    650*
RCVINT  003526R           272     276     280     284     601#
RCVRTN  003604R           613     616#
RCVTBL  000404R           293#    379     417     651     685
RESTRT  002410R           258     432#
RES1    000056R           241#
```

```
RES2    000060R          242#
RPNTR   001766R          338#    441*    616     617*    618     620*
RPNTW   001764R          337#    440*    605*    606*    607     609*
RQ      001624R          299#    337     338     440     441     607     609     618     620
RSTRT   000112R          258#
SBADR   000102R          251#    701*
SCSR    002010R          350#    470*    487
SELECT  001774R          342#    376*    432     434     449     471*    482     523
SELMSK  001772R          341#    448*    449     471     504*    505     522*    523     527*    528
SYLO    000604R          295#    421     645
SOFCNT  000042R          234#
SOFERS= 104406           270#
SOFPAS= 000046R          236#
SPDINT  000032R          230#
SPSIZ = 000040             1#    263
SR1     000016R          223#    381     424     426
SR2     000020R          224#
SR3     000022R          225#
SR4     000024R          226#
START   002054R          229     373#
STAT    000026R          228#
SVR0    000062R          243#
SVR1    000064R          244#
SVR2    000066R          245#
SVR3    000070R          246#
SVR4    000072R          247#
SVR5    000074R          248#
SVR6    000076R          249#
SYSCNT  000052R          238#
TBLSNC  004124R          459     672     680#
TIME    002014R          352#    481     507     563     634     719
TIMER   002016R          353#    481*    507*    518*    563*    634*    719*
TIME1   002020R          354#    508*    516*
TMETBL  002026R          358#    383
TRPDFO= 000022           270#
VCT     002004R          348#    469*    486
VECTOR  000010R          219#    446
WASADR  000104R          253#    703*
WDFR    000116R          260#    375*    414*
WDTO    000114R          259#    374*    413*
XCNT    001700R          305#    393*    589*    595*
XFLAG   000056R          241#
XMITEN  001740R          324#    495*    496     497     586*    588*    674     675
XMITSV  001750R          327#    496*    675*
XMTINT  001722R          274     278     282     286     533#
XMTOK   003356R          556     567#
XMTOUT  003516R          593     597#
XMTRTN  003250R          545     548#
XMTSEL  001730R          319#    495     588
XMTTBL  000304R          291#    377     574     577*    686
XPNTR   001762R          336#    439*    548     549*    550     552*
XPNTW   001760R          335#    438*    537*    538*    539     541*
XQ      001614R          297#    335     336     438     439     539     541     550     552
.       = 004407R        291#    293#    295#    297#    299#    301#
```



```
. ABS.  000000     000
        004407     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDJAH0,XDJAH0/SOL/CRF:SYM=DDXCOM,XDJAH0
 RUN-TIME:  1  2  .3  SECONDS
 RUN-TIME RATIO: 2774=5.7
 CORE USED:  7K  (13 PAGES)
```