DJ11

EXERCISER & ON LINE TEST

MD-11-DZDJB-E

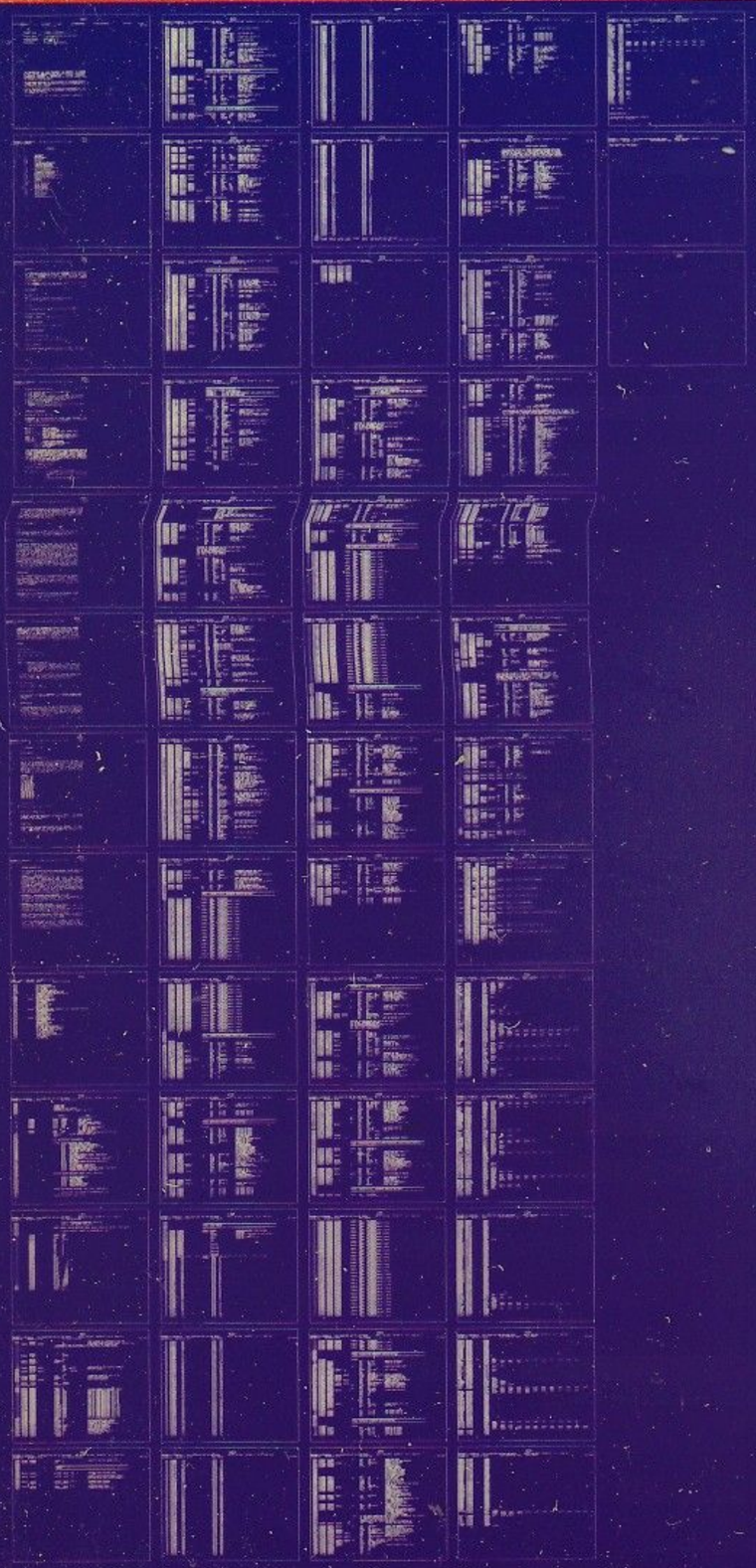
EP-DZDJB-E-DL-B

FICHE 1 OF 1

OCT 1977

digital

MADE IN USA



PRODUCT CODE: MAINDEC-11-DZDJB-E-D

PRODUCT NAME: DJ11 EXERCISER AND ON-LINE TESTS

PROGRAM DATE: MAY 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

CONTENTS

1. ABSTRACT

This program consists of three sub-programs which exrecise the DJ11 Asyncronous Multiplexer. PROGRAM 1 is an off-line exerciser. PROGRAM 2 is an on-line exerciser which continuously transmits the last character received. PROGRAM 3 is an echo test.

NOTE: PROGRAM 1 will run any silo alarm level setting. (For PROGRAM 2 and 3 see Section 9.)

2. REQUIREMENTS

2.1 Equipment

PDP-11 standard computer with console teletype
Up to 16 DJ11 asyncronous multiplexers.

2.2 Storage

This program uses all of 8K, except absolute Loader.

2.3 Preliminary Programs

MAINDEC-11-DZDJA DJ11 Logic Tests

3. LOADING PROCEDURE

Use standard procedure for ABS tapes.

4. STARTING PROCEDURE

4.1 Control Switch Settings

See 5.1 (all down for worst case testing)

4.2 Starting Address

The program should always be started at 200. It may be restarted at 1000 after all parameters have been selected.

4.3 Program and Operator Action

1) Load program into memory using ABS loader.
2) Load address 200.
3) If hardware switch register is available, set switches (see sec. 5.1), all down for worst case, press start.
4) If switch-less processor simply press start.
5) Enter the program number (1, 2, or 3).
6) Select lines if SW<8> is on a 1.
7) PROGRAM 1 will loop and bell will ring once every pass. "EOP" is also printed on each pass.

5. OPERATING PROCEDURE

5.1 Operational Switch Settings

At SA 200, all switches down is worst case testing. For PROGRAM 1 only, the bell will ring and OOeop is printed upon completion of a pass of the entire program.

The switch settings are:

```
SW<15> = 1 ..... HALT ON ERROR
SW<13> = 1 ..... INHIBIT PRINTOUT
SW<12> = 1 ..... PRINT SILO ALARM LEVEL (PROG1 ONLY)
SW<10> = 1 ..... BELL ON ERROR
         0 ..... BELL ON PASS COMPLETE (PROG1 ONLY)
SW<9>  = 1 ..... INHIBIT MAINTENANCE (PROG1 ON-LINE)
SW<8>  = 1 ..... SELECT LINES FOR TEST (SEE 5.3)
PROG1 ONLY:
SW<2:0>= 0 ..... BINARY COUNT PATTERN
         1 ..... "THE QUICK SILVER GRAY FOX ... "
         2 ..... ALPHA-NUMERIC (40-177)
         3-7 ... NOT USED
```

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(I.E.) SWR=XXXXXX NEW=

Possible responses are:

1.	<CR>	If no changes are to be made.
2.	6 DIGITS 0-7	To represent in octal the new switch register value; last digit followed by <CR>.
3.	↑U	To allow reentering value if error is committed keying in swreg value.

Built into the program is the ability to dynamically change the contents of swreg during program execution. By striking ↑G (CNTL G) on console tty the operator sets a request flag to change the contents of swreg, which is processed in key areas of the program code (ie) error routines, after halts end of pass, and other applicable areas.

5.2 Subroutine Abstracts

5.2.1 HLT

This routine (called by an EMT instruction) prints out an error message (see 6.1). To inhibit typeouts, put SW<13> on a 1. To ring the bell on an error, put SW<10> on a 1.

5.2.2 ALMCK (PROG1 only)

in the normal operation the "done" bit is set as each character is read into the fi/fo buffer(silo). but this "done" condition can be delayed to cause done on the 5, 9, or 17 character. this is done by cutting one of the jumpers (w1,w2,w3) on the m7285 control board. the program tests for this "silo alarm level" and if sw12 is set (1) it will print out the level (in octal) at which each dj11 is will set "done". the subroutine also adjusts the character counters to ensure that the maximum number of characters to be transfered is a multiple of the silo alarm level. this ensures that all data will be read out of the silo. done will not set if the number of characters in the fi/fo buffer is less than the silo alarm level. (note character present is set on each character in the buffer, regardless of the silo alarm level.)

5.2.3 TRAPCATCHER

A ".+2" - "HALT" sequence is repeated from 0 - 56 to detect any unexpected traps and a ".+2" - "IOT" sequence is repeated from 60 - 776 to detect any unexpected interrupts. Thus any unexpected traps will HALT at the vector + 2. Any unexpected interrupts will result in an error message and "HALT" in "IOTRAP".

5.3 Program and Operator Action

After the device parameters are reported, the program types "PROGRAM #: ", at which time the operator enters "1", "2", or "3" depending on the sub-program he wishes to run.

If SW<8> is on a 1, the program will TYPE OUT " N SELECT

LINES =" THE OPERATOR RESPONDS BY TYPING IN AN OCTAL NUMBER REPRESENTING THE LINE(S) WHICH ARE TO BE TESTED FOR THAT DJ11.(INPUT A 1 FOR LINE 0,A 7 FOR LINES 0,1,AND 2, ETC. THE SAME AS IF YOU WERE DIRECTLY SETTING THE TCR of the DJ11.) if more dj11's are on the system the n will indicate the next dj11 and the prompt is reissued. when all lines are selected the program will run the selected subprogram.

6. ERRORS

6.1 Error Printout

The format is as follows:

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

Where:

ADR = Address of error HLT
(Rn) = Contents of general register "n". From none to four of these may be typed depending on the number following the HLT; e.g., HLT+3 would type (R1) thru (R3); HLT (by itself) would stop after typing ADR and DJADR.

To find the failing test, look at the listing above the address typed. In most cases the comment beside the HLT tells what was being checked and what was expected.

6.2 Error Recovery

Restart at 200 or 1000.

6.3 Error Counter

An error count is kept in "ERRORS" . IT CAN BE CLEARED FROM THE CONSOLE, BY RESTARTING AT 200, OR BY RELOADING THE PROGRAM.

7. RESTRICTIONS

This program requires that the device addresses follow the floating address convention (DJ11's will be first, starting at 160010, then the DH11's) and that the vector addresses all be contiguous. If the first DJ11 address is nonstandard (other than 160010) then Loc. 1270 must be changed to contain this nonstandard address.

If this program is run with a monitor, i.e. ACT11 or DDP,

only PROGRAM 1 is run.

8. MISCELLANEOUS

8.1 Execution Time (PROG1 only)

Due to the verious baud rates available and the ability to check up to 16 DJ11's at once, the exicution time can vary anywhere from 3 seconds to nearly an hour. The following typical times are for one DJ11 with all lines at the same speed, 8 level code, 2 stop bits, and no parity on a PDP-11/20 . For multiple DJ11's, multiply these times by the number of units selected for test.

BAUD	APPROX RUN TIME
75	00:10:00
110	00:07:00
134.5	00:05:40
150	00:05:00
300	00:02:30
600	00:01:15
1200	00:00:40
1800	00:00:30
2400	00:00:20
4800	00:00:10
9600	00:00:05

8.2 Stack Pointer

Stack is initally set to 1200

8.3 Pass Count (PROG1 only)

A 32 bit (2 words) pass count is kept in "PCNT" . IT CAN BE CLEARED FROM THE CONSOLE, BY RESTARTING AT 200, OR BY RELOADING THE PROGRAM.

8.4 POWER FAIL

EACH PROGRAM CAN BE POWER FAILED WITH NO ERRORS. TO USE, START THE PROGRAM AS USUAL AND POWER DOWN THEN UP AT ANY TIME. THE ROUTINE SHOULD TYPE "POWER" and restart the program with no other error typeouts.

9. PROGRAM DESCRIPTION

This program consists of three sub-programs which exercise the logic of up to 16 DJ11 asynchronous data multiplexers.

PROGRAM 1: EXERCISER (off-line)
This program exercises up to 256 lines (16 DJ11's) simultaneously in maintenance mode. Three different data patterns may be selected from the switch register. The data pattern is repeated a minimum of 16 times for each pass. The program should be run for at least 2 passes with all switches down. SW<9> on a one disables the maintenance mode, requiring turn-around cards at the termination of each line being tested. (Note: The reciever and transmit lines must be jumpered for the same speed.)

PROGRAM 2: CONTINUOUS ECHO EXERCISER (on-line)
This program continuously transmits the last character it received on the respective line. A null (000) will "echo" 72 times and then turn off the transmitter.

PROGRAM 3: ECHO TEST (on-line)
This program transmits a heading (*ECHO TEST*) on each line and then echos everything that it receives.
Caution: If characters are received faster than they can be transmitted, the software buffers may overflow.

Note: The on-line exercisers (PROG2 and PROG3) are operator dependent, and therefore do not loop. I.E. No passes. ACT11 and DDP monitors will only run PROG1.

If the silo alarm level is set for anything other than 1 char. (all 3 jumpers wired in). You must press the key on the terminal, under test, that many times before an echo occures. IE. if the silo alarm level is set for 5 you must press the key 5 times before an echo back.

```
                                .ENABLE ABS
                                .ENABLE AMA
                                ;PROGRAM BY KEN CHAPMAN
                                ;MODIFIED BY DAVID L. ADAMS   MAY 1977

                                ;     SWITCH                    USE
                                ;     ------                    ---------------------
                                ;
                                SW15=   100000          ;HALT ON ERROR
                                SW14=   40000           ;NOT USED
                                SW13=   20000           ;INHIBIT ERROR TYPEOUTS
                                SW12=   10000           ;PRINT SILO ALARM LEVEL
                                SW11=   4000            ;NOT USED
                                SW10=   2000            ;0 - BELL ON PASS COMPLETE
                                                        ;1 - BELL ON ERROR
                                SW9=    1000            ;ON-LINE (PROG1)
                                SW8=    400             ;SELECT LINES (INITIALIZATION TIME ONLY)
                                ;SW<0:2>        SELECT MESSAGE (PROG1 ONLY)
                                .REM!

                                DJ11 REGISTER BIT ASSIGNMENTS:

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

                                RECEIVER BUFFER REGISTER (RBUF) XXXXX2 (READ ONLY)

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

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

                                        BIT0-15 STOP THE SCANNER ON CORRESPONDING LINE

                                TRANSMITTER BUFFER (TBUF) XXXXX6
```

```
                                        BIT0-7  TRANSMITTED CHARACTER (WRITE ONLY)
                                        BIT8-11 LINE NUMBER (READ ONLY)

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

                                        BIT0-15 TRNSMIT A BREAK ON CORRESPONDING LINE!

 2620           104400          SCOPE=  TRAP
  (1)           104000          HLT=    EMT
  (1)           000004          TYPE=   IOT
  (1)           177776          PS=     177776
  (1)           000000          R0=     %0
  (1)           000001          R1=     %1
  (1)           000002          R2=     %2
  (1)           000003          R3=     %3
  (1)           000004          R4=     %4
  (1)           000005          R5=     %5
  (1)           000005          TTY=    %5
  (1)           000006          SP=     %6
  (1)           000007          PC=     %7
  (1)           000007          BELL=   7
  (1)           000001          BIT0=       1
  (1)           000002          BIT1=       2
  (1)           000004          BIT2=       4
  (1)           000010          BIT3=      10
  (1)           000020          BIT4=      20
  (1)           000040          BIT5=      40
  (1)           000100          BIT6=     100
  (1)           000200          BIT7=     200
  (1)           000400          BIT8=     400
  (1)           001000          BIT9=    1000
  (1)           002000          BIT10=   2000
  (1)           004000          BIT11=   4000
  (1)           010000          BIT12=  10000
  (1)           020000          BIT13=  20000
  (1)           040000          BIT14=  40000
  (1)           100000          BIT15= 100000
 2621           000340          LEVEL7 = 340
 2622           000000          OPEN =  0
 2623           001200          STACK = 1200
```

```
2625            000000                  . =     0                       ;TRAP CATCHER IN LOCATIONS 0 THRU 776
 (1)  000000    000000  000000                  0,0                     ;LOCATIONS 0 AND 2 CONTAIN "HALT" INSTRUCTIONS
 (1)                                    ;LOCATIONS 4 THRU 56 CONTAIN ".+2" AND "HALT" IN EVERY VECTOR
 (1)                                    ;LOCATIONS 60 THRU 776 CONTAIN ".+2" AND "IOT" IN EVERY VECTOR
 (1)
 (1)
 (1)
 (1)            000046                  . =     46
 (1)  000046    013574                  $ENDAD
 (1)
 (1)            000174                  . =     174
 (1)  000174    000000                  DISPREG:0
 (1)  000176    000000                  SWREG:  0
 (1)
 (1)            000200                  . =     200
 (1)  000200    000137  005312                  JMP     BEGIN           ;200 ALWAYS IS THE STARTING ADDRESS
2626            001000                  . =     1000                    ;RESTART ADDRESS
2627  001000    000137  006310                  JMP     RESTAR
2628
2629            001200                  . =     1200
2630  001200    000000                  ICNT:   0                       ;ITERATION COUNT-LH, TEST NO.-RH
2631  001202    000000                  ERRORS: 0                       ;ERROR COUNT REGISTER
2632  001204    000000  000000          PCNT:   0,0                     ;PASS COUNT REGISTER
2633
 (1)  001210    177570                  SWR:    177570
 (1)  001212    177570                  DISPLAY:177570
 (1)
2634  001214    000000                  SAVIT:  0
2635  001216    000020                  TIMES:  20                      ;MINIMUM NUMBER OF MESSAGES (PROG1)
2638  001220    000000                  SVSW0:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #0
 (1)  001222    000000                  SVSW1:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #1
 (1)  001224    000000                  SVSW2:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #2
 (1)  001226    000000                  SVSW3:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #3
 (1)  001230    000000                  SVSW4:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #4
 (1)  001232    000000                  SVSW5:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #5
 (1)  001234    000000                  SVSW6:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #6
 (1)  001236    000000                  SVSW7:  OPEN                    ;MAP OF LINES SELECTED, DJ11 #7
 (1)  001240    000000                  SVSW10: OPEN                    ;MAP OF LINES SELECTED, DJ11 #10
 (1)  001242    000000                  SVSW11: OPEN                    ;MAP OF LINES SELECTED, DJ11 #11
 (1)  001244    000000                  SVSW12: OPEN                    ;MAP OF LINES SELECTED, DJ11 #12
 (1)  001246    000000                  SVSW13: OPEN                    ;MAP OF LINES SELECTED, DJ11 #13
 (1)  001250    000000                  SVSW14: OPEN                    ;MAP OF LINES SELECTED, DJ11 #14
 (1)  001252    000000                  SVSW15: OPEN                    ;MAP OF LINES SELECTED, DJ11 #15
 (1)  001254    000000                  SVSW16: OPEN                    ;MAP OF LINES SELECTED, DJ11 #16
 (1)  001256    000000                  SVSW17: OPEN                    ;MAP OF LINES SELECTED, DJ11 #17
2639  001260    000000                  MARK:   0
2640  001262    000030                  BUFSIZ: 30                      ;RECEIVE BUFFER SIZE (PROG3)
2641  001264    000000                  BUFEND: 0                       ;LAST ADDR OF PROG3 BUFFER
2642  001266    000001                  UNITS:  1                       ;NUMBER OF UNITS ON THE SYSTEM
2643  001270    160010                  DEVADR: 160010                  ;FIRST DEVICE ADDRESS
2644  001272    000300                  VECADR: 300                     ;FIRST VECTOR ADDRESS
2645  001274    000240                  RCVLVL: 240                     ;RECEIVER BR LEVEL = 5
2646  001276    000240                  XMTLVL: 240                     ;TRANSMITTER BR LEVEL = 5
2647  001300    000000                  ALMFLG: 0                       ;SILO ALARM LEVEL FLAG
2648  001302    000000                  TIMERA: 0                       ;TIME COUNTERS
2649  001304    000000                  TIMERB: 0
```

```
2650  001306  000000                COUNT:  0                       ;VALUE OF THE SILO ALARM LEVEL
2651  001310  000000                SUM:    0
2652
2653                                ;*****************************************************************************
2654                                ;TABLES
2655                                ;*****************************************************************************
2656
2657  001312  000400                XMTTAB: .BLKW   400             ;TRANSMIT DATA POINTER TABLE
2658
2659  002312  000400                RCVTAB: .BLKW   400             ;RECEIVE DATA POINTER TABLE (PROG1)
2660                                                                ;RECEIVE DATA TABLE (PROG2 AND PROG3)
2661
2662  003312  000400                XMTCNT: .BLKW   400             ;TRANSMIT DATA COUNTER
2663          003313                RCVCNT=XMTCNT+1                 ;RECEIVE DATA COUNTER
2664
2665  004312  000400                MASK:   .BLKW   400             ;CHARACTER MASK TABLE
2666          004313                CNTTAB=MASK+1                   ;ITERATION COUNT AND FLAGS
```

```
                                                    ;SET UP STACK POINTER
005312  012706  001200          BEGIN:  MOV     #STACK, SP
005316  004737  015210                  JSR     PC,SUSWRR
005322  012700  000020                  MOV     #20,R0
005326  012720  014672                  MOV     #IOTRAP,(R0)+   ;IOT VECTOR (20)
005332  012720  000340                  MOV     #340,   (R0)+
005336  012720  014530                  MOV     #PDOWN$,(R0)+   ;POWER FAIL VECTOR (24)
005342  012720  000340                  MOV     #340,   (R0)+
005346  012720  013610                  MOV     #EMT$,  (R0)+   ;EMT VECTOR (30)
005352  012720  000340                  MOV     #340,   (R0)+
005356  005037  001202                  CLR     ERRORS          ;CLEAR ERROR COUNTER
005362  005037  001204                  CLR     PCNT            ;CLEAR PASS COUNTER
005366  005037  001206                  CLR     PCNT+2
005372  012700  000300                  MOV     #300,   R0      ;START OF FLOATING VECTOR AREA
005376  005720                  2$:     TST     (R0)+           ;UPDATE POINTER
005400  010060  177776                  MOV     R0,     -2(R0)  ;PUT ".+2" IN EACH VECTOR
005404  012720  000004                  MOV     #IOT,   (R0)+   ;AND "IOT"
005410  022700  001000                  CMP     #1000,  R0      ;CHECK FOR END OF FLOATING VECTOR AREA
005414  003370                          BGT     2$              ;BRANCH IF MORE
005416  005037  001300                  CLR     ALMFLG          ;CLEAR THE ALARM FLAG
005422  012737  000400  010210          MOV     #256.,CNTNIT    ;SET MAXIMUM SIZE VALUES
005430  012737  000106  010214          MOV     #70.,CNTNIT+4   ;FOR PROG #1
005436  012737  000106  010220          MOV     #70.,CNTNIT+10
005444  012737  000001  001216          MOV     #1,TIMES        ;SET FOR QV ON FIRST PASS

                        ;*****************************************************************
                        ;ROUTINE TO MAP ALL THE DJ11'S ON THE SYSTEM
                        ;*****************************************************************

005452  013700  001270          DJMAP:  MOV     DEVADR, R0      ;GET FIRST FLOATING ADDRESS
005456  012702  000001                  MOV     #1,     R2      ;COUNTER FOR DJ11'S
005462  012737  000002  000006          MOV     #RTI,   @#6     ;RTI WHEN TIME-OUT
005470  005001                  5$:     CLR     R1              ;SET UP COUNTER
005472  000261                  1$:     SEC                     ;SET CARRY
005474  005710                          TST     (R0)            ;CHECK FOR A DEVICE
005476  103404                          BCS     7$              ;BRANCH IF NONE
005500  062700  000010          6$:     ADD     #10,    R0      ;GO TO NEXT DEVICE ADDRESS
005504  005201                          INC     R1              ;COUNT DJ11'S
005506  000771                          BR      1$              ;LOOK FOR MORE

005510  005037  000006          7$:     CLR     @#6             ;RESTORE TIMEOUT VECTOR
005514  010137  001266                  MOV     R1,     UNITS   ;SAVE COUNT
005520  001005                          BNE     GETVEC
005522  000004  015567                  TYPE,   MSG01           ;TYPE "NO DJ11'S!"
005526  000000                          HALT                    ;FATAL ERROR
005530  000137  013522                  JMP     @#DONE          ;RESTART

                        ;*****************************************************************
                        ;ROUTINE TO DETERMINE VECTOR ADDRESSES OF DJ11'S
                        ;*****************************************************************

005534  013746  000020          GETVEC: MOV     @#20,   -(SP)   ;SAVE IOT VECTOR
005540  012737  005570  000020          MOV     #1$,    @#20    ;RESET IOT VECTOR
005546  013701  001270                  MOV     DEVADR, R1      ;FIRST DJ ADDRESS
005552  012711  040400                  MOV     #40400, (R1)    ;SET CSR
                                                                ;BIT8= TRANS SCAN ENABLE
```

```
                                                                ;BIT14= TRANS INTERRUPT ENABLE
005556  012761  000001  000004          MOV     #1,     4(R1)  ;TCR, LINE 0
005564  000001                          WAIT                   ;WAIT FOR AN INTERRUPT
005566  000407                          BR      3$             ;CONTINUE AFTER INTERUPT

005570  011602                  1$:     MOV     (SP),   R2     ;SAVE VECTOR ADR. (+4)
005572  162716  000010                  SUB     #10,    (SP)   ;REPOSITION ADR TO RCV. VEC.
005576  011637  001272                  MOV     (SP),   VECADR ;SAVE FIRST VECTOR
005602  022626                          CMP     (SP)+,  (SP)+  ;RESET STACK FROM IOT
005604  000002                          RTI                    ;RETURN FROM INITIAL INTERUPT

005606  012637  000020          3$:     MOV     (SP)+,  @#20   ;RESTORE IOT VECTOR
005612  005742                          TST     -(R2)          ;POINT TO XMT. VEC. +2
005614  013703  001266                  MOV     UNITS,  R3     ;SET UP UNIT COUNTER

                                ;CHECK THAT VECTORS ARE CONTIGUOUS

005620  005061  000004          2$:     CLR     4(R1)          ;CLEAR TCR
005624  005011                          CLR     (R1)           ;CLEAR CSR
005626  012712  000004                  MOV     #IOT,   (R2)   ;RESTORE IOT TO XMT. VEC.+2
005632  005303                          DEC     R3             ;CHECK FOR MORE DJ11'S
005634  001415                          BEQ     REPORT         ;BRANCH IF DONE
005636  062701  000010                  ADD     #10,    R1     ;UPDATE DJ ADR. POINTER
005642  062702  000010                  ADD     #10,    R2     ;UPDATE VECTOR POINTER
005646  012712  000002                  MOV     #RTI,   (R2)   ;RTI ON INTERRUPT
005652  012711  040400                  MOV     #40400, (R1)   ;SET CSR
005656  012761  000001  000004          MOV     #1,     4(R1)  ;TCR LINE 0
005664  000001                          WAIT                   ;WAIT FOR AN INTERRUPT
005666  000754                          BR      2$

                                ;REPORT CONFIGURATION

005670  032777  020000  173312  REPORT: BIT     #BIT13, @SWR   ;CHECK FOR INHIBIT TYPOUT
005676  001026                          BNE     GETLEN         ;SKIP REPORT IF SET
005700  000004  015402                  TYPE,   MSGMDN
005704  000004  015371                  TYPE,   RETURN
005710  000004  015451                  TYPE,   MSGADR
005714  013705  001270                  MOV     DEVADR,TTY     ;TYPE DEVADR IN OCTAL
(1)     005720  004737  014346          JSR     PC,PRINTR      ;TYPE LEADING ZERO'S
005724  000004  015501                  TYPE,   MSGVEC
005730  013705  001272                  MOV     VECADR,TTY     ;TYPE VECADR IN OCTAL
(1)     005734  004737  014356          JSR     PC,PRINTS      ;AND SUPRESS LEADING ZERO'S
005740  000004  015525                  TYPE,   MSGNUM
005744  013705  001266                  MOV     UNITS,TTY      ;TYPE UNITS IN OCTAL
(1)     005750  004737  014356          JSR     PC,PRINTS      ;AND SUPRESS LEADING ZERO'S
```

```
2779
2780                                    ;*****************************************************************************
2781                                    ;ROUTINE TO MAP CHARACTER LENGTHS
2782                                    ;*****************************************************************************
2783
2784  005754  022737  000176  001210  GETLEN: CMP     #SWREG,SWR
2785  005762  001002                         BNE     6$
2786  005764  004737  015106                 JSR     PC,CNTLU
2787  005770  013700  001266          6$:    MOV     UNITS,  R0      ;SET UP UNIT COUNTER
2788  005774  013701  001270                 MOV     DEVADR, R1      ;SET UP DEVICE ADDRESS POINTER
2789  006000  012702  000001          1$:    MOV     #1,     R2      ;SET UP LINE MARKER
2790  006004  012711  000415                 MOV     #415,   (R1)    ;RCV ENB, CMOS, MAINT., TRANS SCAN ENB
2791  006010  032711  000020          10$:   BIT     #BIT4,  (R1)    ;WAIT FOR MOS TO CLEAR
2792  006014  001375                         BNE     10$
2793  006016  010261  000004          2$:    MOV     R2,     4(R1)   ;TRANS CONTROL, ONE LINE AT A TIME
2794  006022  005711                  3$:    TST     (R1)            ;WAIT FOR TRANS READY
2795  006024  100376                         BPL     3$
2796  006026  012761  000377  000006         MOV     #377,   6(R1)   ;SEND A RUBOUT
2797  006034  006302                         ASL     R2              ;SKIP 4 LINES
2798  006036  006302                         ASL     R2
2799  006040  006302                         ASL     R2
2800  006042  006302                         ASL     R2
2801  006044  103364                         BCC     2$              ;BRANCH BACK IF MORE LINES
2802  006046  005061  000004                 CLR     4(R1)           ;CLEAR TCR
2803  006052  062701  000010                 ADD     #10,    R1      ;UPDATE POINTER TO NEXT UNIT
2804  006056  005300                         DEC     R0              ;CHECK FOR MORE UNITS
2805  006060  001347                         BNE     1$
2806  006062  013700  001266                 MOV     UNITS,  R0      ;SET UP UNIT COUNTER
2807  006066  013701  001270                 MOV     DEVADR, R1      ;SET UP DEVICE ADDRESS POINTER
2808  006072  012702  004312                 MOV     #MASK,  R2      ;SET UP CHAR LEN TABLE POINTER
2809  006076  012703  000004          4$:    MOV     #4,     R3      ;SET UP CHAR COUNTER
2810  006102  016104  000002          5$:    MOV     2(R1),  R4      ;SAVE AND CHECK CHAR PRESENT
2811  006106  100375                         BPL     5$
2812  006110  010405                         MOV     R4,     R5      ;DUP DATA
2813  006112  000305                         SWAB    R5              ;LINE # IN LOW BYTE
2814  006114  042705  177760                 BIC     #177760,R5      ;CLEAR ALL BUT LINE #
2815  006120  006305                         ASL     R5              ;*2
2816  006122  060205                         ADD     R2,     R5      ;MAKE POINTER TO CHAR TABLE
2817  006124  105104                         COMB    R4              ;MAKE DATA INTO MASK
2818  006126  042704  177400                 BIC     #177400,R4      ;CLEAR UPPER BYTE
2819  006132  010425                         MOV     R4,     (R5)+   ;SAVE THE MASK
2820  006134  010425                         MOV     R4,     (R5)+   ;SAVE THE MASK
2821  006136  010425                         MOV     R4,     (R5)+   ;SAVE THE MASK
2822  006140  010425                         MOV     R4,     (R5)+   ;SAVE THE MASK
2823  006142  005303                         DEC     R3              ;COUNT TO 4
2824  006144  001356                         BNE     5$
2825  006146  062701  000010                 ADD     #10,    R1      ;ADDRESS POINTER TO NEXT DJ
2826  006152  062702  000040                 ADD     #40,    R2      ;CHAR LEN TABLE POINTER
2827  006156  005300                         DEC     R0              ;COUNT UNITS
2828  006160  001346                         BNE     4$              ;BRANCH BACK IF MORE
```

```
2833
2834                            ;*****************************************************************************
2835                            ;SELECT THE PROGRAM TO BE RUN
2836                            ;PROGRAM 1:     OFF-LINE EXERCISER
2837                            ;PROGRAM 2:     ON-LINE EXERCISER (TRANSMIT LAST CHARACTER RECEIVED)
2838                            ;PROGRAM 3:     ON-LINE ECHO EXERCISER
2839                            ;*****************************************************************************
2840
2841  006162  005737  000042   SELPRO: TST     @#42            ;CHECK FOR ACT 11 OR DDP
2842  006166  001040                   BNE     ALL             ;BRANCH IF MONITOR
2843  006170  000004  015550           TYPE,   MSGPRG
2844  006174  004537  014002           JSR     R5,     READIN  ;READ A NUMBER FROM THE CTY
2845  006200  006332                   .WORD   PROGNO
2846  006202  001367                   BNE     SELPRO
2847  006204  032777  000400  172776   BIT     #BIT8,  @SWR    ;CHECK FOR SW<8>, SELECT LINES
2848  006212  001426                   BEQ     ALL             ;BRANCH IF NOT
2849  006214  005000                   CLR     R0              ;SET UP UNIT COUNTER, DISPLAY
2850  006216  012701  001220           MOV     #SVSW0,R1       ;SET UP SWITCH TABLE POINTER
2851  006222  000004  015371   SWITCH: TYPE,   RETURN
2852  006226  000004  015346           TYPE,   MNUM
2853  006232  010005                   MOV     R0,TTY          ;PRINT THE NUMBER OF THE DR11C
2854  006234  004737  014356           JSR     PC,PRINTS
2855  006240  000004  015351           TYPE,   MSGSEL          ;ASL FOR THE SELECTED LINES
2856  006244  004537  014002           JSR     R5,READIN
2857  006250  001214                   .WORD   SAVIT
2858  006252  013721  001214           MOV     SAVIT,(R1)+
2859  006256  005200                   INC     R0              ;COUNT UNITS
2860  006260  020037  001266           CMP     R0,     UNITS   ;CHECK FOR MORE UNITS
2861  006264  001356                   BNE     SWITCH          ;BRANCH IF MORE
2862  006266  000410                   BR      RESTAR          ;GO DO IT
2863
2864  006270  013700  001266   ALL:    MOV     UNITS,  R0      ;SET UP UNIT COUNTER
2865  006274  012701  001220           MOV     #SVSW0,R1       ;SET UP SWITCH TABLE POINTER
2866  006300  012721  177777   1$:     MOV     #177777,(R1)+   ;SET ALL LINES
2867  006304  005300                   DEC     R0              ;COUNT UNITS
2868  006306  001374                   BNE     1$              ;BRANCH IF MORE
2869                           .SBTTL          RESTART POINT
2870
2871  006310  013700  006332   RESTAR: MOV     PROGNO, R0
2872  006314  001722                   BEQ     SELPRO
2873  006316  022700  000003           CMP     #3,     R0
2874  006322  103717                   BLO     SELPRO
2875  006324  006300                   ASL     R0
2876  006326  000170  006332           JMP     @PROGAD (R0)
2877
2878  006332                   PROGNO:
2879  006332  000001           PROGAD: 1                       ;DEFAULT TO PROGRAM 1
2880  006334  006342                   PROG1
2881  006336  010762                   PROG2
2882  006340  012166                   PROG3
```

```
                                ;*************************************************************************
                                ;PROGRAM 1:      TRANSMIT AND RECEIVE ALL LINES SIMULTANEOUSLY
                                ;               OFF-LINE:  SW<9> = 0
                                ;               ON-LINE:   SW<9> = 1
                                ;               USES THE DATA TABLE SELECTED BY SW<2:0>
                                ;               EACH LINE REPEATS THE PATTERN AT LEAST 16 TIMES
                                ;               PER PASS.
                                ;*************************************************************************

006342  000005                  PROG1:  RESET                   ;CLEAR OUT THE WORLD
006344  012706  001200                  MOV     #STACK, SP      ;RESET THE STACK POINTER
006350  052737  000340  177776          BIS     #340,@#PS       ;PROCESSOR TO LEVEL 7
006356  012701  001312                  MOV     #XMTTAB,R1      ;FIRST TABLE POINTER
006362  012702  001400                  MOV     #1400,R2        ;LENGTH OF TABLES (WORDS)
006366  005021                  1$:     CLR     (R1)+           ;CLEAR THE TABLE
006370  005302                          DEC     R2
006372  001375                          BNE     1$
006374  012702  000400                  MOV     #400,R2         ;LENGTH OF MASK/COUNT TABLE
006400  005201                  2$:     INC     R1              ;SKIP MASK
006402  105021                          CLRB    (R1)+           ;CLEAR COUNT
006404  005302                          DEC     R2
006406  001374                          BNE     2$

                                ;ROUTINE TO INITIALIZE ALL DJ11'S AND THEIR ISR'S:
                                ;SET UP ALL INTERRUPT VECTORS
                                ;SET UP DEVICE ADDRESSES IN LINKER ROUTINES
                                ;SET CSR'S EVERYTHING ENABLED
                                ;SET TCR'S, ALL LINES ENABLED

006410  005000                  P1INIT: CLR     R0
006412  013701  001270                  MOV     DEVADR, R1
006416  013702  001272                  MOV     VECADR, R2
006422  012703  007354                  MOV     #RISR0, R3
006426  010322                  11$:    MOV     R3,     (R2)+   ;SET UP RECEIVER INTERUPT VECTOR
006430  013722  001274                  MOV     RCVLVL, (R2)+
006434  022323                          CMP     (R3)+,  (R3)+   ;ADD 4 TO R3
006436  010113                          MOV     R1,     (R3)    ;ADDRESS OF CSR
006440  062723  000002                  ADD     #2,     (R3)+   ;ADDRESS OF RBUF
006444  005723                          TST     (R3)+
006446  010322                          MOV     R3,     (R2)+   ;SET UP TRANSMITTER INTERUPT VECTOR
006450  013722  001276                  MOV     XMTLVL, (R2)+
006454  022323                          CMP     (R3)+,  (R3)+
006456  010123                          MOV     R1,     (R3)+   ;ADDRESS OF CSR
006460  005011                          CLR     (R1)            ;CLEAR CSR
006462  052711  050510                  BIS     #50510, (R1)    ;SET UP CSR
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT6 = RECEIVER INTERUPT ENABLE
                                                                ;BIT8 = TRANSMITTER SCAN ENABLE
                                                                ;BIT12 = STATUS ENABLE
                                                                ;BIT14 = TRANSMITTER INTERUPT ENABLE
006466  032711  000020          13$:    BIT     #BIT4,  (R1)    ;CHECK FOR MOS TO CLEAR
006472  001375                          BNE     13$
006474  005737  001300                  TST     ALMFLG          ;HAS THE SILO ALARM LEVEL BEEN CHECKED?
006500  001002                          BNE     10$             ;YES
006502  004737  007006                  JSR     PC,ALMCK        ;NO,GO DO IT
```

```
2943  006506  006300                    10$:    ASL     R0              ;UNIT # * 2
2944  006510  016061  001220  000004            MOV     SVSW0(R0),4(R1) ;SET TCR BITS (CSR + 4)
2945  006516  006200                            ASR     R0              ;RESTORE UNIT COUNTER
2946  006520  012737  000001  001260            MOV     #1,     MARK    ;SET UP MARKER
2947  006526  017705  172456                    MOV     @SWR,   R5      ;GET SWITCH SETTINGS
2948  006532  042705  177770                    BIC     #177770,R5      ;MASK MESSAGE #
2949  006536  006305                            ASL     R5
2950  006540  006305                            ASL     R5
2951  006542  012304                            MOV     (R3)+,  R4      ;SET UP OFFSET TO TABLES
2952  006544  033761  001260  000004  14$:     BIT     MARK,   4(R1)   ;CHECK FOR LINE SELECTED IN TCR
2953  006552  001414                            BEQ     15$
2954  006554  016564  010206  001312            MOV     ADRNIT(5),XMTTAB(4)
2955  006562  016564  010206  002312            MOV     ADRNIT(5),RCVTAB(4)
2956  006570  116564  010210  003312            MOVB    CNTNIT(5),XMTCNT(4)
2957  006576  116564  010210  003313            MOVB    CNTNIT(5),RCVCNT(4)
2958  006604  005724                    15$:    TST     (R4)+           ;INC OFFSET TO NEXT LINE
2959  006606  006337  001260                    ASL     MARK
2960  006612  103354                            BCC     14$
2961  006614  032777  001000  172366            BIT     #BIT9,  @SWR    ;CHECK FOR ON-LINE
2962  006622  001024                            BNE     21$             ;BRANCH IF ON-LINE
2963  006624  052711  000014                    BIS     #14,    (R1)    ;SET THE MAINTENANCE BIT AND CLR MOS
2964  006630  032711  000020            20$:    BIT     #BIT4,(R1)      ;WAIT FOR MOS TO CLEAR
2965  006634  001375                            BNE     20$
2966  006636  052711  000001                    BIS     #1,(R1)         ;TURN ON RCV ENABLE
2967  006642  062701  000010            12$:    ADD     #10,R1          ;POINT TO THE NEXT CSR
2968  006646  005200                            INC     R0
2969  006650  020037  001266                    CMP     R0,     UNITS
2970  006654  001264                            BNE     11$
2971  006656  012737  000001  001300            MOV     #1,ALMFLG       ;SET THE ALARM LEVEL FLAG
2972  006664  042737  000140  177776            BIC     #140,   @#PS    ;LOWER PROCESSOR PRIORITY
2973  006672  000406                            BR      FORGND          ;GO DO IT
2974
2975  006674  052711  000001            21$:    BIS     #1,(R1)         ;TURN ON RCV EN
2976  006700  005761  000002            22$:    TST     2(R1)           ;CLEAR JUNK OUT OF THE RBUF
2977  006704  100775                            BMI     22$
2978  006706  000755                            BR      12$
2982
2983                                  ;*****************************************************************
2984                                  ;PROG1  BACKGROUND PROGRAM TO MONITOR TABLES
2985                                  ;*****************************************************************
2986                                  ; NOTE - PROGRAM MAY HANG IN A LOOP.
2987                                  ; IF THIS HAPPENS, RUN DZDJA.
2988
2989  006710  012701  003312    FORGND: MOV     #XMTCNT,R1
2990  006714  012702  000400            MOV     #400,R2
2991  006720  105711            21$:    TSTB    (R1)            ;CHECK FOR COUNT TABLE CLR
2992  006722  001376                    BNE     21$             ;BRANCH IF NOT
2993  006724  062701  000002            ADD     #2,R1           ;GO TO NEXT LINE ENTRY
2994  006730  005302                    DEC     R2              ;COUNT LINES
2995  006732  001372                    BNE     21$             ;BRANCH IF MORE LINES
2996  006734  012701  003313            MOV     #RCVCNT,R1
2997  006740  012702  000400            MOV     #400,R2
2998  006744  121137  001306    22$:    CMPB    (R1),COUNT      ;IS # OF CHAR LEFT IN RBUF LESS THAN
2999                                                            ;THE SILO ALARM LEVEL
3000  006750  003375                    BGT     22$             ;IF NO WAIT FOR THEM
3001  006752  062701  000002            ADD     #2,R1           ;IF YES IGNORE THEM
```

```
3002  006756  005302                        DEC     R2              ;COUNT LINES
3003  006760  001371                        BNE     22$             ;BRANCH IF MORE LINES
3004  006762  005337  001216                DEC     TIMES           ;DO THIS AGAIN?
3005  006766  001402                        BEQ     23$             ;NO, GET OUT
3006  006770  000137  006342                JMP     PROG1
3007  006774  012737  000020  001216  23$:  MOV     #20,TIMES       ;DO IT 16 TIMES THE NEXT TIME
3008  007002  000137  013522                JMP     @#DONE          ;SKIP ISR'S
3009  007006  012761  000001  000004  ALMCK: MOV     #1,4(R1)        ;SET LINE 0 IN THE TCR
3010  007014  052711  000004                BIS     #BIT2,(R1)      ;SET THE MAINT BIT
3011  007020  052711  000001                BIS     #1,(R1)         ;SET RCV EN AFTER MAINTENANCE BIT
3012  007024  005037  001306                CLR     COUNT
3013  007030  005037  001310                CLR     SUM
3014  007034  005037  001302                CLR     TIMERA
3015  007040  012737  000200  001304  2$:   MOV     #200,TIMERB     ;SET UP TIME CONSTANTS
3016  007046  005711                        TST     (R1)            ;WAIT FOR TRANSFER READY BIT
3017  007050  100373                        BPL     2$
3018  007052  112761  000377  000006        MOVB    #377,6(R1)      ;OUTPUT A CHAR TO TBUF
3019  007060  005237  001306                INC     COUNT           ;COUNT EACH CHAR
3020  007064  105711                1$:     TSTB    (R1)            ;CHECK FOR DONE IN THE CSR
3021  007066  100405                        BMI     3$              ;IF SET GET  OUT OF THE LOOP
3022  007070  004537  007320                JSR     R5,TIME         ;GIVE DONE TIME TO SET
3023  007074  000773                        BR      1$              ;RETURN TO TEST FOR DONE AGAIN
3024  007076  000760                        BR      2$              ;RETURN TO OUTPUT ANOTHER CHAR
3025  007100  000742                        BR      ALMCK           ;ERROR RETURN TRY AGAIN
3026  007102  042711  000001          3$:   BIC     #BIT0,(R1)      ;TURN OFF RCV ENABLE
3027  007106  052711  000010                BIS     #BIT3,(R1)      ;CLEAR MOS
3028  007112  022737  000001  001306        CMP     #1,COUNT        ;IF SILO LEVEL SET FOR 1 THEN GET OUT
3029  007120  001437                        BEQ     4$
3030  007122  063737  001306  001310  5$:   ADD     COUNT,SUM       ;GET THE LARGEST MULTIPLE OF THE
3031  007130  023727  001310  000106        CMP     SUM,#70.        ;SILO ALARM LEVEL AND USE IT IN THE
3032  007136  002771                        BLT     5$              ;THREE SIZE LOCATIONS
3033  007140  001403                        BEQ     6$              ;IF EQUAL USE IT
3034  007142  163737  001306  001310        SUB     COUNT,SUM       ;IF GREATER SUBTRACT ONE COUNT OFF
3035  007150  013737  001310  010214  6$:   MOV     SUM,CNTNIT+4    ;AS CLOSE TO 70 AS POSSIBLE
3036  007156  013737  001310  010220        MOV     SUM,CNTNIT+10
3037  007164  063737  001306  001310  7$:   ADD     COUNT,SUM       ;CONTINUE TO A COUNT OF 256
3038  007172  023727  001310  000377        CMP     SUM,#377
3039  007200  002771                        BLT     7$
3040  007202  163737  001306  001310        SUB     COUNT,SUM
3041  007210  013737  001310  010210  10$:  MOV     SUM,CNTNIT      ;AS CLOSE TO 256 AS POSSIBLE
3042  007216  000411                        BR      11$             ;ALL SET GET OUT
3043  007220  012737  000377  010210  4$:   MOV     #377,CNTNIT     ;PUT SIZE TO MAX VALUE
3044  007226  012737  000106  010214        MOV     #70.,CNTNIT+4
3045  007234  012737  000106  010220        MOV     #70.,CNTNIT+10
3046  007242  042711  000004          11$:  BIC     #BIT2,(R1)      ;TURN OFF THE MAINT BIT
3047  007246  032711  000020          12$:  BIT     #BIT4,(R1)      ;WAIT FOR MOS TO CLEAR
3048  007252  001375                        BNE     12$
3049  007254  004737  015270                JSR     PC,KBDINT       ;GET THE SWITCH REGISTER
3050  007260  032777  010000  171722        BIT     #SW12,@SWR      ;PRINT ALARM LEVEL?
3051  007266  001413                        BEQ     13$             ;NO
3052  007270  000004  015676                TYPE,   MALARM
3053  007274  010105                        MOV     R1,TTY          ;YES, PRINT CSR FIRST -
3054  007276  004737  014346                JSR     PC,PRINTR
3055  007302  000004  015672                TYPE,   MSGDAS
3056  007306  013705  001306                MOV     COUNT,TTY       ;PRINT ALARM LEVEL
3057  007312  004737  014346                JSR     PC,PRINTR
```

```
3058  007316  000207                  13$:    RTS     PC
3059
3060
3061
3062  007320  105237  001302          TIME:   INCB    TIMERA          ;INCREMENT THROUGH ONE WORD
3063  007324  001012                          BNE     1$              ;GO TEST FOR DONE AGAIN
3064  007326  005337  001304                  DEC     TIMERB          ;MAKE TIMERB LARGER IF FAST PROCESSOR
3065  007332  001007                          BNE     1$
3066  007334  023727  001306  000022          CMP     COUNT,#22       ;HAVE OUTPUTTED 18 TIMES
3067  007342  001002                          BNE     2$              ;NO, GO OUTPUT ANOTHER CHAR
3068  007344  104001                          HLT+1                   ;YES,DONE DID NOT SET AFTER 18 OUTPUTS
3069                                                                  ;R1 = CSR
3070  007346  005725                          TST     (R5)+           ;SET R5 FOR ERROR RETURN
3071  007350  005725                  2$:     TST     (R5)+           ;SET R5 FOR NEXT OUTPUT RETURN
3072  007352  000205                  1$:     RTS     R5              ;RETURN FROM ABOVE OR RETEST DONE
3076
3077                                  ;*****************************************************************************
3078                                  ;PROG1  LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
3079                                  ;*****************************************************************************
3080
3086  007354  004037  010070          RISR0:  JSR     R0,RCVISR
 (1)  007360  160012  000000                  .WORD   <160012+<0*10>>,<40*0>
 (1)  007364  004037  007754          XISR0:  JSR     R0,XMTISR
 (1)  007370  160010  000000                  .WORD   <160010+<0*10>>,<40*0>
 (1)  007374  004037  010070          RISR1:  JSR     R0,RCVISR
 (1)  007400  160022  000040                  .WORD   <160012+<1*10>>,<40*1>
 (1)  007404  004037  007754          XISR1:  JSR     R0,XMTISR
 (1)  007410  160020  000040                  .WORD   <160010+<1*10>>,<40*1>
 (1)  007414  004037  010070          RISR2:  JSR     R0,RCVISR
 (1)  007420  160032  000100                  .WORD   <160012+<2*10>>,<40*2>
 (1)  007424  004037  007754          XISR2:  JSR     R0,XMTISR
 (1)  007430  160030  000100                  .WORD   <160010+<2*10>>,<40*2>
 (1)  007434  004037  010070          RISR3:  JSR     R0,RCVISR
 (1)  007440  160042  000140                  .WORD   <160012+<3*10>>,<40*3>
 (1)  007444  004037  007754          XISR3:  JSR     R0,XMTISR
 (1)  007450  160040  000140                  .WORD   <160010+<3*10>>,<40*3>
 (1)  007454  004037  010070          RISR4:  JSR     R0,RCVISR
 (1)  007460  160052  000200                  .WORD   <160012+<4*10>>,<40*4>
 (1)  007464  004037  007754          XISR4:  JSR     R0,XMTISR
 (1)  007470  160050  000200                  .WORD   <160010+<4*10>>,<40*4>
 (1)  007474  004037  010070          RISR5:  JSR     R0,RCVISR
 (1)  007500  160062  000240                  .WORD   <160012+<5*10>>,<40*5>
 (1)  007504  004037  007754          XISR5:  JSR     R0,XMTISR
 (1)  007510  160060  000240                  .WORD   <160010+<5*10>>,<40*5>
 (1)  007514  004037  010070          RISR6:  JSR     R0,RCVISR
 (1)  007520  160072  000300                  .WORD   <160012+<6*10>>,<40*6>
 (1)  007524  004037  007754          XISR6:  JSR     R0,XMTISR
 (1)  007530  160070  000300                  .WORD   <160010+<6*10>>,<40*6>
 (1)  007534  004037  010070          RISR7:  JSR     R0,RCVISR
 (1)  007540  160102  000340                  .WORD   <160012+<7*10>>,<40*7>
 (1)  007544  004037  007754          XISR7:  JSR     R0,XMTISR
 (1)  007550  160100  000340                  .WORD   <160010+<7*10>>,<40*7>
 (1)  007554  004037  010070          RISR10: JSR     R0,RCVISR
 (1)  007560  160112  000400                  .WORD   <160012+<10*10>>,<40*10>
 (1)  007564  004037  007754          XISR10: JSR     R0,XMTISR
 (1)  007570  160110  000400                  .WORD   <160010+<10*10>>,<40*10>
```

```
   (1)  007574  004037  010070          RISR11: JSR     R0,RCVISR
   (1)  007600  160122  000440                  .WORD   <160012+<11*10>>,<40*11>
   (1)  007604  004037  007754          XISR11: JSR     R0,XMTISR
   (1)  007610  160120  000440                  .WORD   <160010+<11*10>>,<40*11>
   (1)  007614  004037  010070          RISR12: JSR     R0,RCVISR
   (1)  007620  160132  000500                  .WORD   <160012+<12*10>>,<40*12>
   (1)  007624  004037  007754          XISR12: JSR     R0,XMTISR
   (1)  007630  160130  000500                  .WORD   <160010+<12*10>>,<40*12>
   (1)  007634  004037  010070          RISR13: JSR     R0,RCVISR
   (1)  007640  160142  000540                  .WORD   <160012+<13*10>>,<40*13>
   (1)  007644  004037  007754          XISR13: JSR     R0,XMTISR
   (1)  007650  160140  000540                  .WORD   <160010+<13*10>>,<40*13>
   (1)  007654  004037  010070          RISR14: JSR     R0,RCVISR
   (1)  007660  160152  000600                  .WORD   <160012+<14*10>>,<40*14>
   (1)  007664  004037  007754          XISR14: JSR     R0,XMTISR
   (1)  007670  160150  000600                  .WORD   <160010+<14*10>>,<40*14>
   (1)  007674  004037  010070          RISR15: JSR     R0,RCVISR
   (1)  007700  160162  000640                  .WORD   <160012+<15*10>>,<40*15>
   (1)  007704  004037  007754          XISR15: JSR     R0,XMTISR
   (1)  007710  160160  000640                  .WORD   <160010+<15*10>>,<40*15>
   (1)  007714  004037  010070          RISR16: JSR     R0,RCVISR
   (1)  007720  160172  000700                  .WORD   <160012+<16*10>>,<40*16>
   (1)  007724  004037  007754          XISR16: JSR     R0,XMTISR
   (1)  007730  160170  000700                  .WORD   <160010+<16*10>>,<40*16>
   (1)  007734  004037  010070          RISR17: JSR     R0,RCVISR
   (1)  007740  160202  000740                  .WORD   <160012+<17*10>>,<40*17>
   (1)  007744  004037  007754          XISR17: JSR     R0,XMTISR
   (1)  007750  160200  000740                  .WORD   <160010+<17*10>>,<40*17>
3090
3091                                    ;*****************************************************************************
3092                                    ;PROG1  TRANSMITTER INTERRUPT SERVICE ROUTINE
3093                                    ;*****************************************************************************
3094
3095  007754                            XMTISR:
   (2)  007754  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
   (2)  007756  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
3096  007760  012001                          MOV     (R0)+,R1
3097  007762  005711                    1$:   TST     (R1)            ;CHECK FOR TRANS READY
3098  007764  100035                          BPL     4$
3099  007766  116102  000007                  MOVB    7(R1),R2        ;GET LINE NO.
3100  007772  006302                          ASL     R2
3101  007774  061002                          ADD     (R0),R2
3102  007776  105762  003312                  TSTB    XMTCNT(2)       ;TST FOR ZERO
3103  010002  001410                          BEQ     2$
3104  010004  117261  001312  000006          MOVB    @XMTTAB(2),6(1) ;SEND A CHARACTER
3105  010012  105362  003312                  DECB    XMTCNT(2)       ;COUNT CHARACTERS
3106  010016  005262  001312                  INC     XMTTAB(2)       ;UPDATE TABLE POINTER
3107  010022  000757                          BR      1$
3108  010024  010346                    2$:   MOV     R3,-(SP)
3109  010026  005062  001312                  CLR     XMTTAB(2)       ;CLEAR TABLE POINTER
3110  010032  161002                          SUB     (R0),R2
3111  010034  006202                          ASR     R2
3112  010036  005003                          CLR     R3
3113  010040  000261                          SEC
3114  010042  006103                    3$:   ROL     R3
3115  010044  005302                          DEC     R2
```

```
3116  010046  100375                        BPL     3$
3117  010050  040361  000004                BIC     R3,4(R1)        ;CLEAR TCR BIT FOR LINE
3118  010054  012603                        MOV     (SP)+,R3        ;RESTORE R3
3119  010056  000741                        BR      1$
3120  010060                          4$:
  (2) 010060  012602                        MOV     (6)+,R2         ;POP STACK INTO R2
  (2) 010062  012601                        MOV     (6)+,R1         ;POP STACK INTO R1
  (2) 010064  012600                        MOV     (6)+,R0         ;POP STACK INTO R0
3121  010066  000002                        RTI
3125
3126                            ;*****************************************************************
3127                            ;PROG1  RECEIVER INTERRUPT SERVICE ROUTINE
3128                            ;*****************************************************************
3129
3130  010070                    RCVISR:
  (2) 010070  010146                        MOV     R1,-(6)         ;PUSH R1 ON STACK
  (2) 010072  010246                        MOV     R2,-(6)         ;PUSH R2 ON STACK
  (2) 010074  010346                        MOV     R3,-(6)         ;PUSH R3 ON STACK
  (2) 010076  010446                        MOV     R4,-(6)         ;PUSH R4 ON STACK
3131  010100  012001                        MOV     (R0)+,R1        ;GET RBUF ADDRESS
3132  010102  011102                1$:     MOV     (R1),R2         ;READ THE DATA
3133  010104  100032                        BPL     7$              ;BRANCH IF NO CHAR PRESENT
3134  010106  032702  070000                BIT     #70000,R2       ;CHECK FOR ERRORS
3135  010112  001403                        BEQ     2$              ;BRANCH IF OK
3136  010114  104002                        HLT+2                   ;RECEIVER ERROR
3137                                                                ;R1=RBUF ADDRESS
3138                                                                ;R2=CONTENTS OF RBUF
3139                                                                ;BIT12=PARITY ERROR
3140                                                                ;BIT13=FRAMING ERROR
3141                                                                ;BIT14=UART OVERRUN
3142  010116  042702  070000                BIC     #70000, R2      ;CLEAR ERROR BITS FOR SPEED
3143  010122  010204                2$:     MOV     R2,     R4      ;DUP THE RBUF
3144  010124  105004                        CLRB    R4              ;CLEAR THE DATA
3145  010126  000304                        SWAB    R4              ;LINE # TO LOW BYTE
3146  010130  106304                        ASLB    R4              ;LINE # * 2, ALSO CLR CHAR PRESENT
3147  010132  061004                        ADD     (R0),R4         ;ADD OFFSET
3148  010134  117403  002312                MOVB    @RCVTAB(R4),R3  ;GET EXPECTED DATA
3149  010140  046403  004312                BIC     MASK(4),R3      ;MASK CHARACTER LENGTH
3150  010144  120302                        CMPB    R3,R2
3151  010146  001403                        BEQ     3$              ;BRANCH IF OK
3152  010150  042703  177400                BIC     #177400,R3      ;MAKE SURE UPPER BYTE CLEAR
3153  010154  104003                        HLT+3                   ;DATA ERROR
3154                                                                ;R1=RBUF ADDRESS
3155                                                                ;R2=CONTENTS OF RBUF(DATA)
3156                                                                ;R3=EXPECTED DATA
3157  010156  105364  003313        3$:     DECB    RCVCNT(4)
3158  010162  001403                        BEQ     7$
3159  010164  005264  002312                INC     RCVTAB(4)       ;UPDATE TABLE POINTER
3160  010170  000744                        BR      1$              ;CONTINUE
3161  010172                          7$:
  (2) 010172  012604                        MOV     (6)+,R4         ;POP STACK INTO R4
  (2) 010174  012603                        MOV     (6)+,R3         ;POP STACK INTO R3
  (2) 010176  012602                        MOV     (6)+,R2         ;POP STACK INTO R2
  (2) 010200  012601                        MOV     (6)+,R1         ;POP STACK INTO R1
  (2) 010202  012600                        MOV     (6)+,R0         ;POP STACK INTO R0
3162  010204  000002                        RTI
```

```
 3166
 3167                                   ;*****************************************************************************
 3168                                   ;PROG1  DATA TABLES
 3169                                   ;*****************************************************************************
 3170
 3171  010206  010252                  ADRNIT: BINARY                  ;SW<2:0>=0 BINARY COUNT PATTERN
 3172  010210  000377                  CNTNIT: 377                     ;SIZE=256.
 3173  010212  010652                          PHRASE                  ;SW<2:0>=1 "THE QUICK SILVER GRAY FOX..."
 3174  010214  000106                          70.                     ;SIZE=70.
 3175  010216  010244                          SIXBIT                  ;SW<2:0>=2 040 THRU 137
 3176  010220  000106                          70.                     ;SIZE=70.
 3177  010222  015734                          END
 3178  010224  000001                          1
 3179  010226  016334                          END+400
 3180  010230  000001                          1
 3181  010232  016734                          END+1000
 3182  010234  000001                          1
 3183  010236  017334                          END+1400
 3184  010240  000001                          1
 3185  010242  017734                          END+2000
 3186  010244  005015  177777  177777  SIXBIT: .ASCII  <15><12><377><377><377><377>    ;CR-LF, FILLERS
 3187  010252     040                  BINARY: .BYTE   40
 3199  010253     041                          .BYTE   41
  (2)  010254     042                          .BYTE   42
  (2)  010255     043                          .BYTE   43
  (2)  010256     044                          .BYTE   44
  (2)  010257     045                          .BYTE   45
  (2)  010260     046                          .BYTE   46
  (2)  010261     047                          .BYTE   47
  (2)  010262     050                          .BYTE   50
  (2)  010263     051                          .BYTE   51
  (2)  010264     052                          .BYTE   52
  (2)  010265     053                          .BYTE   53
  (2)  010266     054                          .BYTE   54
  (2)  010267     055                          .BYTE   55
  (2)  010270     056                          .BYTE   56
  (2)  010271     057                          .BYTE   57
  (2)  010272     060                          .BYTE   60
  (2)  010273     061                          .BYTE   61
  (2)  010274     062                          .BYTE   62
  (2)  010275     063                          .BYTE   63
  (2)  010276     064                          .BYTE   64
  (2)  010277     065                          .BYTE   65
  (2)  010300     066                          .BYTE   66
  (2)  010301     067                          .BYTE   67
  (2)  010302     070                          .BYTE   70
  (2)  010303     071                          .BYTE   71
  (2)  010304     072                          .BYTE   72
  (2)  010305     073                          .BYTE   73
  (2)  010306     074                          .BYTE   74
  (2)  010307     075                          .BYTE   75
  (2)  010310     076                          .BYTE   76
  (2)  010311     077                          .BYTE   77
  (2)  010312     100                          .BYTE   100
  (2)  010313     101                          .BYTE   101
  (2)  010314     102                          .BYTE   102
```

M02


```
(2)  010315  103                    .BYTE  103
(2)  010316  104                    .BYTE  104
(2)  010317  105                    .BYTE  105
(2)  010320  106                    .BYTE  106
(2)  010321  107                    .BYTE  107
(2)  010322  110                    .BYTE  110
(2)  010323  111                    .BYTE  111
(2)  010324  112                    .BYTE  112
(2)  010325  113                    .BYTE  113
(2)  010326  114                    .BYTE  114
(2)  010327  115                    .BYTE  115
(2)  010330  116                    .BYTE  116
(2)  010331  117                    .BYTE  117
(2)  010332  120                    .BYTE  120
(2)  010333  121                    .BYTE  121
(2)  010334  122                    .BYTE  122
(2)  010335  123                    .BYTE  123
(2)  010336  124                    .BYTE  124
(2)  010337  125                    .BYTE  125
(2)  010340  126                    .BYTE  126
(2)  010341  127                    .BYTE  127
(2)  010342  130                    .BYTE  130
(2)  010343  131                    .BYTE  131
(2)  010344  132                    .BYTE  132
(2)  010345  133                    .BYTE  133
(2)  010346  134                    .BYTE  134
(2)  010347  135                    .BYTE  135
(2)  010350  136                    .BYTE  136
(2)  010351  137                    .BYTE  137
(2)  010352  140                    .BYTE  140
(2)  010353  141                    .BYTE  141
(2)  010354  142                    .BYTE  142
(2)  010355  143                    .BYTE  143
(2)  010356  144                    .BYTE  144
(2)  010357  145                    .BYTE  145
(2)  010360  146                    .BYTE  146
(2)  010361  147                    .BYTE  147
(2)  010362  150                    .BYTE  150
(2)  010363  151                    .BYTE  151
(2)  010364  152                    .BYTE  152
(2)  010365  153                    .BYTE  153
(2)  010366  154                    .BYTE  154
(2)  010367  155                    .BYTE  155
(2)  010370  156                    .BYTE  156
(2)  010371  157                    .BYTE  157
(2)  010372  160                    .BYTE  160
(2)  010373  161                    .BYTE  161
(2)  010374  162                    .BYTE  162
(2)  010375  163                    .BYTE  163
(2)  010376  164                    .BYTE  164
(2)  010377  165                    .BYTE  165
(2)  010400  166                    .BYTE  166
(2)  010401  167                    .BYTE  167
(2)  010402  170                    .BYTE  170
(2)  010403  171                    .BYTE  171
(2)  010404  172                    .BYTE  172
```

```
(2)  010405      173                    .BYTE   173
(2)  010406      174                    .BYTE   174
(2)  010407      175                    .BYTE   175
(2)  010410      176                    .BYTE   176
(2)  010411      177                    .BYTE   177
(2)  010412      200                    .BYTE   200
(2)  010413      201                    .BYTE   201
(2)  010414      202                    .BYTE   202
(2)  010415      203                    .BYTE   203
(2)  010416      204                    .BYTE   204
(2)  010417      205                    .BYTE   205
(2)  010420      206                    .BYTE   206
(2)  010421      207                    .BYTE   207
(2)  010422      210                    .BYTE   210
(2)  010423      211                    .BYTE   211
(2)  010424      212                    .BYTE   212
(2)  010425      213                    .BYTE   213
(2)  010426      214                    .BYTE   214
(2)  010427      215                    .BYTE   215
(2)  010430      216                    .BYTE   216
(2)  010431      217                    .BYTE   217
(2)  010432      220                    .BYTE   220
(2)  010433      221                    .BYTE   221
(2)  010434      222                    .BYTE   222
(2)  010435      223                    .BYTE   223
(2)  010436      224                    .BYTE   224
(2)  010437      225                    .BYTE   225
(2)  010440      226                    .BYTE   226
(2)  010441      227                    .BYTE   227
(2)  010442      230                    .BYTE   230
(2)  010443      231                    .BYTE   231
(2)  010444      232                    .BYTE   232
(2)  010445      233                    .BYTE   233
(2)  010446      234                    .BYTE   234
(2)  010447      235                    .BYTE   235
(2)  010450      236                    .BYTE   236
(2)  010451      237                    .BYTE   237
(2)  010452      240                    .BYTE   240
(2)  010453      241                    .BYTE   241
(2)  010454      242                    .BYTE   242
(2)  010455      243                    .BYTE   243
(2)  010456      244                    .BYTE   244
(2)  010457      245                    .BYTE   245
(2)  010460      246                    .BYTE   246
(2)  010461      247                    .BYTE   247
(2)  010462      250                    .BYTE   250
(2)  010463      251                    .BYTE   251
(2)  010464      252                    .BYTE   252
(2)  010465      253                    .BYTE   253
(2)  010466      254                    .BYTE   254
(2)  010467      255                    .BYTE   255
(2)  010470      256                    .BYTE   256
(2)  010471      257                    .BYTE   257
(2)  010472      260                    .BYTE   260
(2)  010473      261                    .BYTE   261
(2)  010474      262                    .BYTE   262
```

```
(2)  010475      263                    .BYTE   263
(2)  010476      264                    .BYTE   264
(2)  010477      265                    .BYTE   265
(2)  010500      266                    .BYTE   266
(2)  010501      267                    .BYTE   267
(2)  010502      270                    .BYTE   270
(2)  010503      271                    .BYTE   271
(2)  010504      272                    .BYTE   272
(2)  010505      273                    .BYTE   273
(2)  010506      274                    .BYTE   274
(2)  010507      275                    .BYTE   275
(2)  010510      276                    .BYTE   276
(2)  010511      277                    .BYTE   277
(2)  010512      300                    .BYTE   300
(2)  010513      301                    .BYTE   301
(2)  010514      302                    .BYTE   302
(2)  010515      303                    .BYTE   303
(2)  010516      304                    .BYTE   304
(2)  010517      305                    .BYTE   305
(2)  010520      306                    .BYTE   306
(2)  010521      307                    .BYTE   307
(2)  010522      310                    .BYTE   310
(2)  010523      311                    .BYTE   311
(2)  010524      312                    .BYTE   312
(2)  010525      313                    .BYTE   313
(2)  010526      314                    .BYTE   314
(2)  010527      315                    .BYTE   315
(2)  010530      316                    .BYTE   316
(2)  010531      317                    .BYTE   317
(2)  010532      320                    .BYTE   320
(2)  010533      321                    .BYTE   321
(2)  010534      322                    .BYTE   322
(2)  010535      323                    .BYTE   323
(2)  010536      324                    .BYTE   324
(2)  010537      325                    .BYTE   325
(2)  010540      326                    .BYTE   326
(2)  010541      327                    .BYTE   327
(2)  010542      330                    .BYTE   330
(2)  010543      331                    .BYTE   331
(2)  010544      332                    .BYTE   332
(2)  010545      333                    .BYTE   333
(2)  010546      334                    .BYTE   334
(2)  010547      335                    .BYTE   335
(2)  010550      336                    .BYTE   336
(2)  010551      337                    .BYTE   337
(2)  010552      340                    .BYTE   340
(2)  010553      341                    .BYTE   341
(2)  010554      342                    .BYTE   342
(2)  010555      343                    .BYTE   343
(2)  010556      344                    .BYTE   344
(2)  010557      345                    .BYTE   345
(2)  010560      346                    .BYTE   346
(2)  010561      347                    .BYTE   347
(2)  010562      350                    .BYTE   350
(2)  010563      351                    .BYTE   351
(2)  010564      352                    .BYTE   352
```

```
   (2)  010565     353                                .BYTE   353
   (2)  010566     354                                .BYTE   354
   (2)  010567     355                                .BYTE   355
   (2)  010570     356                                .BYTE   356
   (2)  010571     357                                .BYTE   357
   (2)  010572     360                                .BYTE   360
   (2)  010573     361                                .BYTE   361
   (2)  010574     362                                .BYTE   362
   (2)  010575     363                                .BYTE   363
   (2)  010576     364                                .BYTE   364
   (2)  010577     365                                .BYTE   365
   (2)  010600     366                                .BYTE   366
   (2)  010601     367                                .BYTE   367
   (2)  010602     370                                .BYTE   370
   (2)  010603     371                                .BYTE   371
   (2)  010604     372                                .BYTE   372
   (2)  010605     373                                .BYTE   373
   (2)  010606     374                                .BYTE   374
   (2)  010607     375                                .BYTE   375
   (2)  010610     376                                .BYTE   376
   (2)  010611     377                                .BYTE   377
3205    010612     000                                .BYTE   0
   (2)  010613     001                                .BYTE   1
   (2)  010614     002                                .BYTE   2
   (2)  010615     003                                .BYTE   3
   (2)  010616     004                                .BYTE   4
   (2)  010617     005                                .BYTE   5
   (2)  010620     006                                .BYTE   6
   (2)  010621     007                                .BYTE   7
   (2)  010622     010                                .BYTE   10
   (2)  010623     011                                .BYTE   11
   (2)  010624     012                                .BYTE   12
   (2)  010625     013                                .BYTE   13
   (2)  010626     014                                .BYTE   14
   (2)  010627     015                                .BYTE   15
   (2)  010630     016                                .BYTE   16
   (2)  010631     017                                .BYTE   17
   (2)  010632     020                                .BYTE   20
   (2)  010633     021                                .BYTE   21
   (2)  010634     022                                .BYTE   22
   (2)  010635     023                                .BYTE   23
   (2)  010636     024                                .BYTE   24
   (2)  010637     025                                .BYTE   25
   (2)  010640     026                                .BYTE   26
   (2)  010641     027                                .BYTE   27
   (2)  010642     030                                .BYTE   30
   (2)  010643     031                                .BYTE   31
   (2)  010644     032                                .BYTE   32
   (2)  010645     033                                .BYTE   33
   (2)  010646     034                                .BYTE   34
   (2)  010647     035                                .BYTE   35
   (2)  010650     036                                .BYTE   36
   (2)  010651     037                                .BYTE   37
3206
3207    010652  005015  177777  177777  PHRASE:       .ASCII  <15><12><377><377><377><377>
3208    010660  044124  020105  052521                .ASCIZ  "THE QUICK SILVER GRAY FOX JUMPED OVER 9,876,543,210.0 LAZY DOGS!"
```

```
        010666  041511  020113  044523
        010674  053114  051105  043440
        010702  040522  020131  047506
        010710  020130  052512  050115
        010716  042105  047440  042526
        010724  020122  026071  033470
        010732  026066  032065  026063
        010740  030462  027060  020060
        010746  040514  054532  042040
        010754  043517  020523     000
 3209           010762                 .EVEN
```

```
                        ;*************************************************************************
                        ;PROGRAM 2:     ON-LINE MULTI-ECHO EXERCISER
                        ;               TRANSMITS THE LAST CHARACTER RECEIVED ON ITS RESPECTIVE
                        ;               LINE.  A CARRIAGE RETURN AND LINE FEED ARE INSERTED
                        ;               EVERY 72 CHARACTERS.
                        ;*************************************************************************

010762  000005          PROG2:  RESET                   ;CLEAR OUT THE WORLD
010764  012706  001200          MOV     #STACK, SP      ;RESET THE STACK POINTER
010770  052737  000340  177776  BIS     #340,   @#PS    ;PROCESSOR TO LEVEL 7
010776  012701  001312          MOV     #XMTTAB,R1      ;FIRST TABLE POINTER
011002  012702  001400          MOV     #1400,  R2              ;LENGTH OF TABLES (WORDS)
011006  005021          1$:     CLR     (R1)+           ;CLEAR THE TABLE
011010  005302                  DEC     R2
011012  001375                  BNE     1$
011014  012702  000400          MOV     #400,R2         ;LENGTH OF MASK/COUNT TABLE
011020  005201          2$:     INC     R1              ;SKIP MASK
011022  105021                  CLRB    (R1)+           ;CLEAR COUNT
011024  005302                  DEC     R2
011026  001374                  BNE     2$

                        ;ROUTINE TO INITIALIZE ALL DJ11'S AND THEIR ISR'S:
                        ;SET UP ALL INTERRUPT VECTORS
                        ;SET UP DEVICE ADDRESSES IN LINKER ROUTINES
                        ;SET CSR'S EVERYTHING ENABLED
                        ;SET TCR'S, ALL LINES ENABLED

011030  005000          P2INIT: CLR     R0
011032  013701  001270          MOV     DEVADR, R1
011036  013702  001272          MOV     VECADR, R2
011042  012703  011220          MOV     #R2SR0,R3       ;SET UP POINTER TO LINKERS
011046  010322          1$:     MOV     R3,     (R2)+   ;SET UP RECEIVER INTERUPT VECTOR
011050  013722  001274          MOV     RCVLVL, (R2)+
011054  022323                  CMP     (R3)+,  (R3)+   ;ADD 4 TO R3
011056  010113                  MOV     R1,     (R3)    ;ADDRESS OF CSR
011060  062723  000002          ADD     #2,     (R3)+   ;ADDRESS OF RBUF
011064  005723                  TST     (R3)+
011066  010322                  MOV     R3,     (R2)+   ;SET UP TRANSMITTER INTERUPT VECTOR
011070  013722  001276          MOV     XMTLVL, (R2)+
011074  022323                  CMP     (R3)+,  (R3)+
011076  010123                  MOV     R1,     (R3)+   ;ADDRESS OF CSR
011100  012721  050501          MOV     #50501, (R1)+   ;SET UP CSR
                                                        ;BIT0 = RECEIVER ENABLE
                                                        ;BIT6 = RECEIVER INTERUPT ENABLE
                                                        ;BIT8 = TRANSMITTER SCAN ENABLE
                                                        ;BIT12 = STATUS ENABLE
                                                        ;BIT14 = TRANSMITTER INTERUPT ENABLE
011104  005721                  TST     (R1)+
011106  006300                  ASL     R0              ;UNIT # * 2
011110  016011  001220          MOV     SVSW0(R0),(R1)  ;SET TCR BITS FOR SELECTED LINES
011114  006200                  ASR     R0              ;RESET UNIT COUNTER
011116  012737  000001  001260  MOV     #1,     MARK    ;SET UP MARKER
011124  012304                  MOV     (R3)+,  R4      ;SET UP OFFSET TO TABLES
011126  033711  001260  4$:     BIT     MARK,   (R1)    ;CHECK FOR LINE SELECTED
011132  001406                  BEQ     5$
```

```
3270  011134  012764  015604  001312          MOV     #MSGP2, XMTTAB(4) ;SET UP XMTR TABLE
3271  011142  012764  000045  003312          MOV     #45,    XMTCNT(4) ;SET UP COUNT
3272  011150  005724                  5$:     TST     (R4)+             ;INC OFFSET TO NEXT LINE
3273  011152  006337  001260                  ASL     MARK
3274  011156  103363                          BCC     4$
3275  011160  022121                          CMP     (R1)+,  (R1)+     ;ADD 4
3276  011162  005200                          INC     R0
3277  011164  020037  001266                  CMP     R0,     UNITS
3278  011170  001326                          BNE     1$
3279  011172  042737  000140  177776          BIC     #140,   @#PS      ;LOWER PROCESSOR PRIORITY
3283
3284                                  ;*************************************************************************
3285                                  ;PROG2  FOREGROUND PROGRAM TO READ/WRITE MEMORY
3286                                  ;*************************************************************************
3287
3288  011200  012700  020000          FORP2:  MOV     #20000,R0         ;TOP OF 4K BANK OF MEMORY
3289  011204  000241                          CLC
3290  011206  005540                  1$:     ADC     -(R0)             ;FAST READ/WRITE TO MEMORY
3291  011210  001376                          BNE     1$                ;RAPID REPEAT
3292  011212  005700                          TST     R0                ;CHECK FOR LOC 0
3293  011214  001374                          BNE     1$                ;BRANCH IF MORE MEMORY
3294  011216  000770                          BR      FORP2             ;LOOP FOR EVER!
3298
3299                                  ;*************************************************************************
3300                                  ;PROG2 LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
3301                                  ;*************************************************************************
3302
3308  011220  004037  012012          R2SR0:  JSR     R0,P2RISR
 (1)  011224  160012  000000                  .WORD   <160012+<0*10>>,<0*40>
 (1)  011230  004037  011620          X2SR0:  JSR     R0,P2XISR
 (1)  011234  160020  000000                  .WORD   <160020+<0*10>>,<0*40>
 (1)  011240  004037  012012          R2SR1:  JSR     R0,P2RISR
 (1)  011244  160022  000040                  .WORD   <160012+<1*10>>,<1*40>
 (1)  011250  004037  011620          X2SR1:  JSR     R0,P2XISR
 (1)  011254  160030  000040                  .WORD   <160020+<1*10>>,<1*40>
 (1)  011260  004037  012012          R2SR2:  JSR     R0,P2RISR
 (1)  011264  160032  000100                  .WORD   <160012+<2*10>>,<2*40>
 (1)  011270  004037  011620          X2SR2:  JSR     R0,P2XISR
 (1)  011274  160040  000100                  .WORD   <160020+<2*10>>,<2*40>
 (1)  011300  004037  012012          R2SR3:  JSR     R0,P2RISR
 (1)  011304  160042  000140                  .WORD   <160012+<3*10>>,<3*40>
 (1)  011310  004037  011620          X2SR3:  JSR     R0,P2XISR
 (1)  011314  160050  000140                  .WORD   <160020+<3*10>>,<3*40>
 (1)  011320  004037  012012          R2SR4:  JSR     R0,P2RISR
 (1)  011324  160052  000200                  .WORD   <160012+<4*10>>,<4*40>
 (1)  011330  004037  011620          X2SR4:  JSR     R0,P2XISR
 (1)  011334  160060  000200                  .WORD   <160020+<4*10>>,<4*40>
 (1)  011340  004037  012012          R2SR5:  JSR     R0,P2RISR
 (1)  011344  160062  000240                  .WORD   <160012+<5*10>>,<5*40>
 (1)  011350  004037  011620          X2SR5:  JSR     R0,P2XISR
 (1)  011354  160070  000240                  .WORD   <160020+<5*10>>,<5*40>
 (1)  011360  004037  012012          R2SR6:  JSR     R0,P2RISR
 (1)  011364  160072  000300                  .WORD   <160012+<6*10>>,<6*40>
 (1)  011370  004037  011620          X2SR6:  JSR     R0,P2XISR
 (1)  011374  160100  000300                  .WORD   <160020+<6*10>>,<6*40>
 (1)  011400  004037  012012          R2SR7:  JSR     R0,P2RISR
```

```
  (1)  011404  160102  000340                .WORD   <160012+<7*10>>,<7*40>
  (1)  011410  004037  011620      X2SR7:    JSR     R0,P2XISR
  (1)  011414  160110  000340                .WORD   <160020+<7*10>>,<7*40>
  (1)  011420  004037  012012      R2SR10:   JSR     R0,P2RISR
  (1)  011424  160112  000400                .WORD   <160012+<10*10>>,<10*40>
  (1)  011430  004037  011620      X2SR10:   JSR     R0,P2XISR
  (1)  011434  160120  000400                .WORD   <160020+<10*10>>,<10*40>
  (1)  011440  004037  012012      R2SR11:   JSR     R0,P2RISR
  (1)  011444  160122  000440                .WORD   <160012+<11*10>>,<11*40>
  (1)  011450  004037  011620      X2SR11:   JSR     R0,P2XISR
  (1)  011454  160130  000440                .WORD   <160020+<11*10>>,<11*40>
  (1)  011460  004037  012012      R2SR12:   JSR     R0,P2RISR
  (1)  011464  160132  000500                .WORD   <160012+<12*10>>,<12*40>
  (1)  011470  004037  011620      X2SR12:   JSR     R0,P2XISR
  (1)  011474  160140  000500                .WORD   <160020+<12*10>>,<12*40>
  (1)  011500  004037  012012      R2SR13:   JSR     R0,P2RISR
  (1)  011504  160142  000540                .WORD   <160012+<13*10>>,<13*40>
  (1)  011510  004037  011620      X2SR13:   JSR     R0,P2XISR
  (1)  011514  160150  000540                .WORD   <160020+<13*10>>,<13*40>
  (1)  011520  004037  012012      R2SR14:   JSR     R0,P2RISR
  (1)  011524  160152  000600                .WORD   <160012+<14*10>>,<14*40>
  (1)  011530  004037  011620      X2SR14:   JSR     R0,P2XISR
  (1)  011534  160160  000600                .WORD   <160020+<14*10>>,<14*40>
  (1)  011540  004037  012012      R2SR15:   JSR     R0,P2RISR
  (1)  011544  160162  000640                .WORD   <160012+<15*10>>,<15*40>
  (1)  011550  004037  011620      X2SR15:   JSR     R0,P2XISR
  (1)  011554  160170  000640                .WORD   <160020+<15*10>>,<15*40>
  (1)  011560  004037  012012      R2SR16:   JSR     R0,P2RISR
  (1)  011564  160172  000700                .WORD   <160012+<16*10>>,<16*40>
  (1)  011570  004037  011620      X2SR16:   JSR     R0,P2XISR
  (1)  011574  160200  000700                .WORD   <160020+<16*10>>,<16*40>
  (1)  011600  004037  012012      R2SR17:   JSR     R0,P2RISR
  (1)  011604  160202  000740                .WORD   <160012+<17*10>>,<17*40>
  (1)  011610  004037  011620      X2SR17:   JSR     R0,P2XISR
  (1)  011614  160210  000740                .WORD   <160020+<17*10>>,<17*40>
3312
3313                                ;*****************************************************************************
3314                                ;PROG2  TRANSMITTER INTERRUPT SERVICE ROUTINE
3315                                ;*****************************************************************************
3316
3317  011620                        P2XISR:
  (2)  011620  010146                       MOV     R1,-(6)           ;PUSH R1 ON STACK
  (2)  011622  010246                       MOV     R2,-(6)           ;PUSH R2 ON STACK
3318  011624  012001                       MOV     (R0)+,R1
3319  011626  005711                1$:    TST     (R1)              ;CHECK FOR TRANS READY
3320  011630  100064                       BPL     4$
3321  011632  116102  000007              MOVB    7(R1),R2          ;GET LINE NO.
3322  011636  006302                       ASL     R2
3323  011640  061002                       ADD     (R0),R2
3324  011642  105762  003312              TSTB    XMTCNT(2)         ;TST FOR ZERO
3325  011646  001413                       BEQ     2$                ;GET OUT
3326  011650  117261  001312  000006      MOVB    @XMTTAB(2),6(R1);SEND A CHARACTER
3327  011656  105362  003312              DECB    XMTCNT(2)         ;COUNT CHARACTERS
3328  011662  105762  004313              TSTB    CNTTAB(2)         ;CHECK FOR MESSAGE OR DATA
3329  011666  001357                       BNE     1$                ;BRANCH IF DATA
3330  011670  005262  001312              INC     XMTTAB(2)         ;UPDATE TABLE POINTER
```

```
3331  011674  000754                          BR      1$
3332  011676  105162  004313          2$:     COMB    CNTTAB(2)               ;CHANGE FLAG
3333  011702  001430                          BEQ     3$                      ;BRANCH IF WAS DATA
3334  011704  012762  002312  001312          MOV     #RCVTAB,XMTTAB(2) ;SET UP POINTER TO RECEIVER TABLE
3335  011712  060262  001312                  ADD     R2,     XMTTAB(2) ;ADD OFFSET
3336  011716  112762  000110  003312          MOVB    #72.,   XMTCNT(2) ;COUNT 72. CHARACTERS TO THE LINE
3337  011724  105762  002312                  TSTB    RCVTAB(2)               ;CHECK FOR A BREAK
3338  011730  001336                          BNE     1$                      ;BRANCH IF REAL DATA
3339  011732  161002                          SUB     (R0),   R2              ;RECOVER LINE NUMBER
3340  011734  006202                          ASR     R2
3341  011736  005037  001260                  CLR     MARK                    ;SET UP MARKER
3342  011742  000261                          SEC
3343  011744  006137  001260          5$:     ROL     MARK                    ;MOVE MARKER
3344  011750  005302                          DEC     R2                      ;COUNT LINES
3345  011752  100374                          BPL     5$                      ;BRANCH IF MORE
3346  011754  043761  001260  000004          BIC     MARK,   4(R1)   ;CLEAR TCR BIT
3347  011762  000721                          BR      1$                      ;CONTINUE
3348  011764  012762  015371  001312  3$:     MOV     #RETURN,XMTTAB(2) ;TYPE CARRIAGE RETURN, LINE FEED
3349  011772  112762  000002  003312          MOVB    #2,     XMTCNT(2) ;COUNTER OF 2 CHARACTERS
3350  012000  000712                          BR      1$
3351  012002                          4$:
  (2) 012002  012602                          MOV     (6)+,R2                 ;POP STACK INTO R2
  (2) 012004  012601                          MOV     (6)+,R1                 ;POP STACK INTO R1
  (2) 012006  012600                          MOV     (6)+,R0                 ;POP STACK INTO R0
3352  012010  000002                          RTI
3356
3357                                  ;*****************************************************************************
3358                                  ;PROG2  RECEIVER INTERRUPT SERVICE ROUTINE
3359                                  ;*****************************************************************************
3360
3361  012012                          P2RISR:
  (2) 012012  010146                          MOV     R1,-(6)                 ;PUSH R1 ON STACK
  (2) 012014  010246                          MOV     R2,-(6)                 ;PUSH R2 ON STACK
  (2) 012016  010346                          MOV     R3,-(6)                 ;PUSH R3 ON STACK
3362  012020  012001                          MOV     (R0)+,R1                ;GET RBUF ADDRESS
3363  012022  011102          1$:     MOV     (R1),R2                 ;READ THE DATA
3364  012024  100053                          BPL     7$                      ;BRANCH IF NO CHAR PRESENT
3365  012026  032702  070000                  BIT     #70000,R2               ;CHECK FOR ERRORS
3366  012032  001402                          BEQ     2$                      ;BRANCH IF OK
3367  012034  104002                          HLT+2                           ;RECEIVER ERROR
3368                                                                          ;R1=RBUF ADDRESS
3369                                                                          ;R2=CONTENTS OF RBUF
3370                                                                          ;BIT12=PARITY ERROR
3371                                                                          ;BIT13=FRAMING ERROR
3372                                                                          ;BIT14=UART OVERRUN
3373  012036  000771                          BR      1$                      ;FORGET THE DATA
3374
3375  012040  010203          2$:     MOV     R2,     R3              ;DUP THE RBUF
3376  012042  105003                          CLRB    R3                      ;CLEAR THE DATA
3377  012044  000303                          SWAB    R3                      ;LINE # TO LOW BYTE
3378  012046  106303                          ASLB    R3                      ;LINE # * 2, ALSO CLR CHAR PRESENT
3379  012050  061003                          ADD     (R0),R3                 ;ADD OFFSET
3380  012052  136302  004312                  BITB    MASK(3),R2              ;CHECK CHARACTER LENGTH
3381  012056  001401                          BEQ     3$                      ;BRANCH IF OK
3382  012060  104002                          HLT+2                           ;CHARACTER LENGTH ERROR
3383                                                                          ;R1=RBUF ADDRESS
```

```
3384                                                    ;R2=CONTENTS OF RBUF(DATA)
3385  012062  105763  002312          3$:   TSTB   RCVTAB(3)          ;CHECK FOR BREAK
3386  012066  001017                       BNE    5$                 ;BRANCH IF REAL DATA
3387  012070  110263  002312               MOVB   R2,     RCVTAB(3) ;SAVE THE DATA
3388  012074  161003                       SUB    (R0),   R3         ;RECOVER LINE NUMBER
3389  012076  006203                       ASR    R3
3390  012100  005037  001260               CLR    MARK               ;SET UP MARKER
3391  012104  000261                       SEC
3392  012106  006137  001260          4$:   ROL    MARK               ;UPDATE MARKER
3393  012112  005303                       DEC    R3                 ;COUNT LINES
3394  012114  100374                       BPL    4$                 ;BRANCH IF MORE
3395  012116  053761  001260  000002       BIS    MARK,   2(R1)      ;SET TCR BIT
3396  012124  000736                       BR     1$                 ;CONTINUE
3397
3398  012126  110263  002312          5$:   MOVB   R2,     RCVTAB(3) ;SAVE THE DATA
3399  012132  105163  004313               COMB   CNTTAB(3)          ;SET MESSAGE FLAG
3400  012136  012763  015371  001312       MOV    #RETURN,XMTTAB(3) ;TYPE CARRIAGE RETURN, LINE FEED
3401  012144  112763  000002  003312       MOVB   #2,     XMTCNT(3) ;MESSAGE LENGTH
3402  012152  000723                       BR     1$
3403  012154                          7$:
  (2) 012154  012603                       MOV    (6)+,R3            ;POP STACK INTO R3
  (2) 012156  012602                       MOV    (6)+,R2            ;POP STACK INTO R2
  (2) 012160  012601                       MOV    (6)+,R1            ;POP STACK INTO R1
  (2) 012162  012600                       MOV    (6)+,R0            ;POP STACK INTO R0
3404  012164  000002                       RTI
```

```
3409
3410                                    ;*****************************************************************************
3411                                    ;PROGRAM 3:     ECHO EXERCISER
3412                                    ;*****************************************************************************
3413
3414  012166  000005                   PROG3:  RESET                   ;CLEAR OUT THE WORLD
3415  012170  012706  001200                   MOV     #STACK, SP      ;RESET THE STACK POINTER
3416  012174  052737  000340  177776           BIS     #340,   @#PS    ;PROCESSOR TO LEVEL 7
3417  012202  012701  001312                   MOV     #XMTTAB,R1      ;FIRST TABLE POINTER
3418  012206  012702  001400                   MOV     #1400,  R2      ;LENGTH OF TABLES (WORDS)
3419  012212  005021                   1$:     CLR     (R1)+           ;CLEAR THE TABLE
3420  012214  005302                           DEC     R2
3421  012216  001375                           BNE     1$
3422  012220  012702  000400                   MOV     #400,R2         ;LENGTH OF MASK/COUNT TABLE
3423  012224  005201                   2$:     INC     R1              ;SKIP MASK
3424  012226  105021                           CLRB    (R1)+           ;CLEAR COUNT
3425  012230  005302                           DEC     R2
3426  012232  001374                           BNE     2$
3427  012234  012737  015734  001264           MOV     #END,BUFEND     ;GENERATE LAST BUFFER ADDRESS
3428  012242  063737  001262  001264           ADD     BUFSIZ,BUFEND
3429  012250  012705  015734                   MOV     #END,   R5      ;SET UP BUFFER POINTER
3430
3431                                    ;ROUTINE TO INITIALIZE ALL DJ11'S AND THEIR ISR'S:
3432                                    ;SET UP ALL INTERRUPT VECTORS
3433                                    ;SET UP DEVICE ADDRESSES IN LINKER ROUTINES
3434                                    ;SET CSR'S EVERYTHING ENABLED
3435                                    ;SET TCR'S, ALL LINES ENABLED
3436
3437  012254  005000                   P3INIT: CLR     R0
3438  012256  013701  001270                   MOV     DEVADR, R1
3439  012262  013702  001272                   MOV     VECADR, R2
3440  012266  012703  012550                   MOV     #R3SR0,R3       ;SET UP POINTER TO LINKERS
3441  012272  010322                   1$:     MOV     R3,     (R2)+   ;SET UP RECEIVER INTERUPT VECTOR
3442  012274  013722  001274                   MOV     RCVLVL, (R2)+
3443  012300  022323                           CMP     (R3)+,  (R3)+   ;ADD 4 TO R3
3444  012302  010113                           MOV     R1,     (R3)    ;ADDRESS OF CSR
3445  012304  062723  000002                   ADD     #2,     (R3)+   ;ADDRESS OF RBUF
3446  012310  005723                           TST     (R3)+
3447  012312  010322                           MOV     R3,     (R2)+   ;SET UP TRANSMITTER INTERUPT VECTOR
3448  012314  013722  001276                   MOV     XMTLVL, (R2)+
3449  012320  022323                           CMP     (R3)+,  (R3)+
3450  012322  010123                           MOV     R1,     (R3)+   ;ADDRESS OF CSR
3451  012324  012721  050400                   MOV     #50400, (R1)+   ;SET UP CSR, TRANSMITTER ONLY
3452                                                                    ;BIT8 = TRANSMITTER SCAN ENABLE
3453                                                                    ;BIT12 = STATUS ENABLE
3454                                                                    ;BIT14 = TRANSMITTER INTERUPT ENABLE
3455  012330  005721                           TST     (R1)+
3456  012332  006300                           ASL     R0              ;UNIT # * 2
3457  012334  016011  001220                   MOV     SVSW0(R0),(R1)  ;SET TCR BITS FOR SELECTED LINES
3458  012340  006200                           ASR     R0              ;RESET UNIT COUNTER
3459  012342  012737  000001  001260           MOV     #1,     MARK    ;SET UP MARKER
3460  012350  012304                           MOV     (R3)+,  R4      ;SET UP OFFSET TO TABLES
3461  012352  010246                           MOV     R2,-(6)         ;PUSH R2 ON STACK
 (2)  012354  010346                           MOV     R3,-(6)         ;PUSH R3 ON STACK
3462  012356  033711  001260           2$:     BIT     MARK,   (R1)    ;CHECK FOR LINE SELECTED
3463  012362  001420                           BEQ     6$
```

```
3464  012364  010564  001312              MOV     R5, XMTTAB(4)   ;SET UP HEADER MESSAGE
3465  012370  010564  002312              MOV     R5, RCVTAB(4)   ;SET UP RECEIVER TABLE
3466  012374  013702  001262              MOV     BUFSIZ, R2      ;SET UP COUNTER
3467  012400  012703  015652              MOV     #MSGP3, R3      ;SET UP MESSAGE POINTER
3468  012404  112325              3$:     MOVB    (R3)+,  (R5)+   ;MOVE MESSAGE INTO BUFFER
3469  012406  001404                      BEQ     5$              ;BRANCH IF END OF MESSAGE
3470  012410  005302                      DEC     R2              ;COUNT BUFFER SIZE
3471  012412  001374                      BNE     3$              ;BRANCH IF MORE
3472  012414  000403                      BR      6$              ;BRANCH IF DONE
3473  012416  105025              4$:     CLRB    (R5)+           ;CLEAR REST OF BUFFER
3474  012420  005302              5$:     DEC     R2              ;COUNT BUFFER SIZE
3475  012422  001375                      BNE     4$              ;BRANCH IF MORE
3476  012424  005724              6$:     TST     (R4)+           ;INC OFFSET TO NEXT LINE
3477  012426  006337  001260              ASL     MARK
3478  012432  103351                      BCC     2$
3479  012434  012603                      MOV     (6)+,R3         ;POP STACK INTO R3
  (2) 012436  012602                      MOV     (6)+,R2         ;POP STACK INTO R2
3480  012440  022121                      CMP     (R1)+,  (R1)+   ;ADD 4
3481  012442  005200                      INC     R0
3482  012444  020037  001266              CMP     R0,     UNITS
3483  012450  001310                      BNE     1$
3484  012452  042737  000140  177776      BIC     #140,   @#PS    ;LOWER PROCESSOR PRIORITY
3488
3489                                  ;*****************************************************************************
3490                                  ;PROG3  FOREGROUND PROGRAM TO START RECEIVERS, THEN EXERCISE MEMORY.
3491                                  ;*****************************************************************************
3492
3493  012460  012701  001312      FORP3:  MOV     #XMTTAB,R1
3494  012464  012702  000400              MOV     #400,R2
3495  012470  005711              1$:     TST     (R1)            ;CHECK FOR XMTR TABLE CLR
3496  012472  001376                      BNE     1$              ;BRANCH IF NOT
3497  012474  062701  000002              ADD     #2,R1           ;GO TO NEXT LINE ENTRY
3498  012500  005302                      DEC     R2              ;COUNT LINES
3499  012502  001372                      BNE     1$              ;BRANCH IF MORE LINES
3500  012504  013700  001266              MOV     UNITS,  R0      ;SET UP UNIT COUNTER
3501  012510  013701  001270              MOV     DEVADR, R1      ;AND DEVICE ADDRESS POINTER
3502  012514  052711  000101      2$:     BIS     #101,   (R1)    ;SET RECEIVER ENABLES OF CSR
3503                                                              ;BIT0 = RECEIVER ENABLE
3504                                                              ;BIT6 = RCV INTERRUPT ENABLE
3505  012520  062701  000010              ADD     #10,    R1      ;UPDATE TO NEXT DJ11
3506  012524  005300                      DEC     R0              ;COUNT DJ11'S
3507  012526  001372                      BNE     2$
3508  012530  012700  020000      MEMX3:  MOV     #20000,R0       ;TOP OF 4K BANK OF MEMORY
3509  012534  000241                      CLC
3510  012536  005540              1$:     ADC     -(R0)           ;FAST READ/WRITE TO MEMORY
3511  012540  001376                      BNE     1$              ;RAPID REPEAT
3512  012542  005700                      TST     R0              ;CHECK FOR LOC 0
3513  012544  001374                      BNE     1$              ;BRANCH IF MORE MEMORY
3514  012546  000770                      BR      MEMX3           ;LOOP FOR EVER!
3518
3519                                  ;*****************************************************************************
3520                                  ;PROG3  LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
3521                                  ;*****************************************************************************
3522
3528  012550  004037  013300      R3SR0:  JSR     R0,P3RISR
  (1) 012554  160012  000000              .WORD   <160012+<0*10>>,<0*40>
```

```
(1)  012560  004037  013150  X3SR0:  JSR     R0,P3XISR
(1)  012564  160020  000000          .WORD   <160020+<0*10>>,<0*40>
(1)  012570  004037  013300  R3SR1:  JSR     R0,P3RISR
(1)  012574  160022  000040          .WORD   <160012+<1*10>>,<1*40>
(1)  012600  004037  013150  X3SR1:  JSR     R0,P3XISR
(1)  012604  160030  000040          .WORD   <160020+<1*10>>,<1*40>
(1)  012610  004037  013300  R3SR2:  JSR     R0,P3RISR
(1)  012614  160032  000100          .WORD   <160012+<2*10>>,<2*40>
(1)  012620  004037  013150  X3SR2:  JSR     R0,P3XISR
(1)  012624  160040  000100          .WORD   <160020+<2*10>>,<2*40>
(1)  012630  004037  013300  R3SR3:  JSR     R0,P3RISR
(1)  012634  160042  000140          .WORD   <160012+<3*10>>,<3*40>
(1)  012640  004037  013150  X3SR3:  JSR     R0,P3XISR
(1)  012644  160050  000140          .WORD   <160020+<3*10>>,<3*40>
(1)  012650  004037  013300  R3SR4:  JSR     R0,P3RISR
(1)  012654  160052  000200          .WORD   <160012+<4*10>>,<4*40>
(1)  012660  004037  013150  X3SR4:  JSR     R0,P3XISR
(1)  012664  160060  000200          .WORD   <160020+<4*10>>,<4*40>
(1)  012670  004037  013300  R3SR5:  JSR     R0,P3RISR
(1)  012674  160062  000240          .WORD   <160012+<5*10>>,<5*40>
(1)  012700  004037  013150  X3SR5:  JSR     R0,P3XISR
(1)  012704  160070  000240          .WORD   <160020+<5*10>>,<5*40>
(1)  012710  004037  013300  R3SR6:  JSR     R0,P3RISR
(1)  012714  160072  000300          .WORD   <160012+<6*10>>,<6*40>
(1)  012720  004037  013150  X3SR6:  JSR     R0,P3XISR
(1)  012724  160100  000300          .WORD   <160020+<6*10>>,<6*40>
(1)  012730  004037  013300  R3SR7:  JSR     R0,P3RISR
(1)  012734  160102  000340          .WORD   <160012+<7*10>>,<7*40>
(1)  012740  004037  013150  X3SR7:  JSR     R0,P3XISR
(1)  012744  160110  000340          .WORD   <160020+<7*10>>,<7*40>
(1)  012750  004037  013300  R3SR10: JSR     R0,P3RISR
(1)  012754  160112  000400          .WORD   <160012+<10*10>>,<10*40>
(1)  012760  004037  013150  X3SR10: JSR     R0,P3XISR
(1)  012764  160120  000400          .WORD   <160020+<10*10>>,<10*40>
(1)  012770  004037  013300  R3SR11: JSR     R0,P3RISR
(1)  012774  160122  000440          .WORD   <160012+<11*10>>,<11*40>
(1)  013000  004037  013150  X3SR11: JSR     R0,P3XISR
(1)  013004  160130  000440          .WORD   <160020+<11*10>>,<11*40>
(1)  013010  004037  013300  R3SR12: JSR     R0,P3RISR
(1)  013014  160132  000500          .WORD   <160012+<12*10>>,<12*40>
(1)  013020  004037  013150  X3SR12: JSR     R0,P3XISR
(1)  013024  160140  000500          .WORD   <160020+<12*10>>,<12*40>
(1)  013030  004037  013300  R3SR13: JSR     R0,P3RISR
(1)  013034  160142  000540          .WORD   <160012+<13*10>>,<13*40>
(1)  013040  004037  013150  X3SR13: JSR     R0,P3XISR
(1)  013044  160150  000540          .WORD   <160020+<13*10>>,<13*40>
(1)  013050  004037  013300  R3SR14: JSR     R0,P3RISR
(1)  013054  160152  000600          .WORD   <160012+<14*10>>,<14*40>
(1)  013060  004037  013150  X3SR14: JSR     R0,P3XISR
(1)  013064  160160  000600          .WORD   <160020+<14*10>>,<14*40>
(1)  013070  004037  013300  R3SR15: JSR     R0,P3RISR
(1)  013074  160162  000640          .WORD   <160012+<15*10>>,<15*40>
(1)  013100  004037  013150  X3SR15: JSR     R0,P3XISR
(1)  013104  160170  000640          .WORD   <160020+<15*10>>,<15*40>
(1)  013110  004037  013300  R3SR16: JSR     R0,P3RISR
(1)  013114  160172  000700          .WORD   <160012+<16*10>>,<16*40>
```

```
  (1)  013120  004037  013150          X3SR16: JSR     R0,P3XISR
  (1)  013124  160200  000700                  .WORD   <160020+<16*10>>,<16*40>
  (1)  013130  004037  013300          R3SR17: JSR     R0,P3RISR
  (1)  013134  160202  000740                  .WORD   <160012+<17*10>>,<17*40>
  (1)  013140  004037  013150          X3SR17: JSR     R0,P3XISR
  (1)  013144  160210  000740                  .WORD   <160020+<17*10>>,<17*40>
3532
3533                                   ;*****************************************************************************
3534                                   ;PROG3  TRANSMITTER INTERRUPT SERVICE ROUTINE
3535                                   ;*****************************************************************************
3536
3537  013150                           P3XISR:
  (2)  013150  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
  (2)  013152  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
3538  013154  012001                          MOV     (R0)+,R1
3539  013156  005711                  1$:     TST     (R1)            ;CHECK FOR TRANS READY
3540  013160  100043                          BPL     4$
3541  013162  116102  000007                 MOVB    7(R1),R2        ;GET LINE NO.
3542  013166  006302                          ASL     R2
3543  013170  061002                          ADD     (R0),R2
3544  013172  117261  001312  000006         MOVB    @XMTTAB(2),6(R1);SEND A CHARACTER
3545  013200  105072  001312                 CLRB    @XMTTAB(2)      ;CLR TABLE AFTER USE
3546  013204  005262  001312                 INC     XMTTAB(2)       ;UPDATE TABLE POINTER
3547  013210  023762  001264  001312         CMP     BUFEND,XMTTAB(2) ;CHECK FOR END OF BUFFER
3548  013216  001003                          BNE     5$              ;BRANCH IF NOT
3549  013220  012762  015734  001312         MOV     #END,XMTTAB(2) ;RESET BUFFER POINTER
3550  013226  105772  001312          5$:     TSTB    @XMTTAB(2)      ;CHECK NEXT CHARACTER
3551  013232  001351                          BNE     1$              ;BRANCH IF MORE DATA
3552  013234  010346                  2$:     MOV     R3,-(SP)
3553  013236  005062  001312                 CLR     XMTTAB(2)       ;CLEAR TABLE POINTER
3554  013242  161002                          SUB     (R0),R2
3555  013244  006202                          ASR     R2
3556  013246  005003                          CLR     R3
3557  013250  000261                          SEC
3558  013252  006103                  3$:     ROL     R3
3559  013254  005302                          DEC     R2
3560  013256  100375                          BPL     3$
3561  013260  040361  000004                 BIC     R3,4(R1)        ;CLEAR TCR BIT FOR LINE
3562  013264  012603                          MOV     (SP)+,R3        ;RESTORE R3
3563  013266  000733                          BR      1$
3564  013270                           4$:
  (2)  013270  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
  (2)  013272  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
  (2)  013274  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
3565  013276  000002                          RTI
3569
3570                                   ;*****************************************************************************
3571                                   ;PROG3  RECEIVER INTERRUPT SERVICE ROUTINE
3572                                   ;*****************************************************************************
3573
3574  013300                           P3RISR:
  (2)  013300  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
  (2)  013302  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
  (2)  013304  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
  (2)  013306  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
3575  013310  012001                          MOV     (R0)+,R1        ;GET RBUF ADDRESS
```

```
3576  013312  011102                  1$:     MOV     (R1),R2         ;READ THE DATA
3577  013314  100074                          BPL     8$              ;BRANCH IF NO CHAR PRESENT
3578  013316  032702  070000                  BIT     #70000,R2       ;CHECK FOR ERRORS
3579  013322  001402                          BEQ     2$              ;BRANCH IF OK
3580  013324  104002                          HLT+2                   ;RECEIVER ERROR
3581                                                                  ;R1=RBUF ADDRESS
3582                                                                  ;R2=CONTENTS OF RBUF
3583                                                                  ;BIT12=PARITY ERROR
3584                                                                  ;BIT13=FRAMING ERROR
3585                                                                  ;BIT14=UART OVERRUN
3586  013326  000771                          BR      1$              ;SKIP BAD DATA
3587  013330  010204                  2$:     MOV     R2,     R4      ;DUP THE RBUF
3588  013332  105004                          CLRB    R4              ;CLEAR THE DATA
3589  013334  000304                          SWAB    R4              ;LINE # TO LOW BYTE
3590  013336  106304                          ASLB    R4              ;LINE # * 2, ALSO CLR CHAR PRESENT
3591  013340  061004                          ADD     (R0),R4         ;ADD OFFSET
3592  013342  136402  004312                BITB    MASK(4),R2      ;CHECK CHARACTER LENGTH
3593  013346  001401                          BEQ     3$              ;BRANCH IF OK
3594  013350  104002                          HLT+2                   ;CHARACTER LENGTH ERROR
3595                                                                  ;R1=RBUF ADDRESS
3596                                                                  ;R2=CONTENTS OF RBUF(DATA)
3597  013352  005764  002312          3$:     TST     RCVTAB(4)       ;CHECK FOR UNSELECTED LINE
3598  013356  001002                          BNE     4$              ;BRANCH IF OK
3599  013360  104002                          HLT+2                   ;RECEIVED DATA ON UNSELECTED LINE
3600                                                                  ;R1 = RBUF ADDRESS
3601                                                                  ;R2 = CONTENTS OF RBUF
3602  013362  000753                          BR      1$              ;IGNORE THE DATA
3603  013364  105774  002312          4$:     TSTB    @RCVTAB(4)      ;CHECK FOR DATA BUFFER FULL
3604  013370  001403                          BEQ     5$              ;BRANCH IF OK
3605  013372  104002                          HLT+2                   ;SOFTWARE DATA BUFFER OVERFLOW
3606                                                                  ;POSSIBLE TRANSMITTER PROBLEM
3607                                                                  ;R1 = RBUF ADDRESS
3608                                                                  ;R2 = CONTENTS OF RBUF
3609                                  ;NOTE:  IF THE ABOVE ERROR WAS DUE TO OVERLOAD, INCREASING THE CONTENTS
3610                                  ;       OF "BUFSIZ" MAY RECTIFY THE PROBLEM.
3611                                  ;       "BUFSIZ" MUST BE A MULTIPLE OF 2.
3612                                  ;       INCREASING IT MAY CAUSE THE BUFFERS TO OVERFLOW 4K.
3613  013374  000137  012166                  JMP     PROG3           ;RESTART ON THIS TYPE ERROR
3614
3615  013400  005764  001312          5$:     TST     XMTTAB(4)       ;CHECK FOR TRANSMITTER ACTIVE
3616  013404  001414                          BEQ     6$              ;BRANCH IF INACTIVE
3617  013406  110274  002312                  MOVB    R2, @RCVTAB(4)  ;PUT THE DATA IN THE BUFFER
3618  013412  005264  002312                  INC     RCVTAB(4)       ;UPDATE POINTER TO NEXT SPACE
3619  013416  023764  001264  002312          CMP     BUFEND,RCVTAB(4) ;CHECK FOR END OF BUFFER
3620  013424  001332                          BNE     1$              ;BRANCH IF NOT
3621  013426  012764  015734  002312          MOV     #END,RCVTAB(4) ;RESET BUFFER POINTER
3622  013434  000726                          BR      1$
3623  013436  012764  015734  002312  6$:     MOV     #END,   RCVTAB(4) ;RESET TABLE POINTER
3624  013444  016464  002312  001312          MOV     RCVTAB(4),XMTTAB(4)
3625  013452  110274  002312                  MOVB    R2, @RCVTAB(4)
3626  013456  005264  002312                  INC     RCVTAB(4)       ;UPDATE POINTER TO NEXT SPACE
3627  013462  161004                          SUB     (R0),R4
3628  013464  006204                          ASR     R4
3629  013466  005003                          CLR     R3
3630  013470  000261                          SEC
3631  013472  006103                  7$:     ROL     R3
```

```
 3632  013474  005304                        DEC     R4
 3633  013476  100375                        BPL     7$
 3634  013500  050361  000002                BIS     R3,2(R1)        ;SET TCR BIT FOR LINE
 3635  013504  000702                        BR      1$
 3636  013506                          8$:
   (2) 013506  012604                        MOV     (6)+,R4         ;POP STACK INTO R4
   (2) 013510  012603                        MOV     (6)+,R3         ;POP STACK INTO R3
   (2) 013512  012602                        MOV     (6)+,R2         ;POP STACK INTO R2
   (2) 013514  012601                        MOV     (6)+,R1         ;POP STACK INTO R1
   (2) 013516  012600                        MOV     (6)+,R0         ;POP STACK INTO R0
 3637  013520  000002                        RTI
 3638  013522                          DONE:
   (1) 013522  004737  015270                JSR     PC,     KBDINT
   (1) 013526  062737  000001  001206        ADD     #1,PCNT+2       ;ADD 1 TO THE PASS COUNT
   (1) 013534  005537  001204                ADC     PCNT            ;MAKE IT DOUBLE PREC.
   (1) 013540  000004  015340                TYPE    ,MEOP           ;END OF PASS INDICATOR
   (1) 013544  032777  002000  165436        BIT     #SW10,@SWR      ;RING THE BELL?
   (1) 013552  001004                        BNE     4$              ;NO!
   (1) 013554  000004  000007                TYPE    ,BELL           ;RING THE BELL
   (1) 013560  000004  000177                TYPE    ,177            ;TYPE A FILLER FOR 11/05
   (1) 013564  013700  000042          4$:   MOV     @#42,R0         ;GET MONITOR ADDRESS
   (1) 013570  001405                        BEQ     3$              ;IF NONE
   (1) 013572  000005                        RESET                   ;RESET AND
   (1)         013574                        $ENDAD = .
   (1) 013574  004710                        JSR     7,(0)           ;GO TO MONITOR
   (1) 013576  000240                        NOP                     ;SAVE ROOM
   (1) 013600  000240                        NOP                     ;FOR
   (1) 013602  000240                        NOP                     ;ACT11
   (1) 013604  000137  006310          3$:   JMP     RESTAR          ;RETURN
   (1)
```

```
3640
3641                                    ;       $HLT            ERROR TYPEOUT HANDLER
 (1)
 (1)                                    ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
 (1)                                    ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
 (1)                                    ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
 (1)                                    ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUEMENT
 (1)                                    ;(HLT+3) WILL BE PLACED IN "HLTCT$:" FOR ADITIONAL TYPEOUTS.
 (1)
 (1)  013610  004737  015270            EMT$:   JSR     PC,     KBDINT
 (1)  013614  032777  002000  165366            BIT     #SW10,@SWR      ;BELL ON ERROR?
 (1)  013622  001402                            BEQ     1$              ;NO - SKIP
 (1)  013624  000004  000007                    TYPE    ,BELL           ;RING BELL
 (1)  013630  005237  001202            1$:     INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
 (1)  013634  032777  020000  165346            BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
 (1)  013642  001026                            BNE     2$              ;SKIP TYPEOUTS
 (2)  013644  000004  013650                    TYPE    .+2             ;.ASCIZ <15><12>
 (1)  013654  011637  013740                    MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
 (1)  013660  162737  000002  013740            SUB     #2,HLTADR       ;FUDGE ADDRESS
 (1)  013666  117737  000046  013736            MOVB    @HLTADR,HLTCT$  ;GET HLT ARGUEMENT
 (2)  013674  013705  013740                    MOV     HLTADR,TTY      ;TYPE HLTADR IN OCTAL
 (2)  013700  004737  014346                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
 (2)  013704  000004  013710                    TYPE    .+2             ;.ASCIZ " "
 (1)  013714  004737  013742                    JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
 (1)  013720  005777  165264            2$:     TST     @SWR            ;HALT ON ERROR
 (1)  013724  100001                            BPL     .+4             ;SKIP IF CONTINUE
 (1)  013726  000000                            HALT                    ;HALT ON ERROR!
 (1)  013730  004737  015270                    JSR     PC,KBDINT
 (1)  013734  000002                            RTI                     ;RETURN
 (1)
 (1)  013736  000000            HLTCT$: 0                               ;HLT ARGUEMENT
 (1)  013740  000000            HLTADR: 0                               ;LAST HLT INSTRUCTION EXECUTED
3642
3643  013742  042737  007700  013764  ERROR$: BIC     #7700,2$
3644  013750  105337  013736          1$:     DECB    HLTCT$
3645  013754  100411                          BMI     3$
3646  013756  062737  000100  013764          ADD     #100,2$
3647  013764  010005                  2$:     MOV     %0,TTY          ;TYPE REGISTER X IN OCTAL
3648  013766  004737  014346                  JSR     %7,PRINTR
3649  013772  000004  015377                  TYPE,   SPACE
3650  013776  000764                          BR      1$
3651  014000  000207                  3$:     RTS     PC
```

```
                                ;SUBROUTINE TO SAVE INPUT AS OCTAL NUMBER
                                ;
014002  012737  000001  014274  READIN: MOV     #1,INHRE
014010  004737  014150                  JSR     PC,     READ$   ;GO READ TTY UNTIL CR
014014  005037  014274                  CLR     INHRE
014020  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
(2) 014022  010246                      MOV     R2,-(6)         ;PUSH R2 ON STACK
(2) 014024  010346                      MOV     R3,-(6)         ;PUSH R3 ON STACK
014026  012501                          MOV     (R5)+,  R1
014030  012737  000020  015166          MOV     #20,CNT
014036  012702  014276                  MOV     #INPUT,R2
014042  122712  000120                  CMPB    #120,(R2)       ;CHECK FOR "P"
014046  001425                          BEQ     3$
014050  005011                          CLR     (R1)
014052  112203                  1$:     MOVB    (R2)+,R3
014054  120327  000015                  CMPB    R3,#15
014060  001420                          BEQ     3$              ;BRANCH WHEN DONE
014062  162703  000060                  SUB     #60,R3
014066  032703  177770                  BIT     #177770,R3
014072  001013                          BNE     3$              ;BRANCH IF BAD DATA
014074  006311                          ASL     (R1)
014076  103410                          BCS     2$
014100  006311                          ASL     (R1)
014102  103406                          BCS     2$
014104  006311                          ASL     (R1)
014106  103404                          BCS     2$
014110  050311                          BIS     R3,(R1)
014112  005337  015166                  DEC     CNT
014116  000755                          BR      1$
014120  000244                  2$:     CLZ                     ;MAKE SURE Z-BIT IS CLR
014122  013737  177776  014146  3$:     MOV     @#PS,   PSTEMP  ;SAVE CONDITION CODES
014130  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
(2) 014132  012602                      MOV     (6)+,R2         ;POP STACK INTO R2
(2) 014134  012601                      MOV     (6)+,R1         ;POP STACK INTO R1
014136  013737  014146  177776          MOV     PSTEMP, @#PS    ;RESTORE CONDITION CODES
014144  000205                          RTS     R5
                                ;
014146  000000          PSTEMP: 0               ;TEMPORARY STORAGE FOR PS
                                ;
014150  010346          READ$:  MOV     R3,-(6)         ;SAVE R3
(1) 014152  012703  014276      1$:     MOV     #INPUT,R3       ;GET ADDRESS
(1) 014156  022703  014316      2$:     CMP     #INPUT+20,R3    ;BUFFER FULL?
(1) 014162  001415                      BEQ     4$              ;YES - TYPE "?"
(1) 014164  105737  177560              TSTB    @#177560        ;WAIT FOR
(1) 014170  100375                      BPL     .-4             ;A CHARACTER
(1) 014172  113713  177562              MOVB    @#177562,(3)    ;GET CHARACTER
(1) 014176  142713  000200              BICB    #200,(3)        ;GET RID OF JUNK
(1) 014202  122713  000177              CMPB    #177,(3)        ;IS IT A RUBOUT
(1) 014206  001403                      BEQ     4$              ;SKIP IF NOT
(1) 014210  122713  000025              CMPB    #25,(3)
(1) 014214  001006                      BNE     3$
(1) 014216                      4$:
(2) 014216  000004  014222              TYPE    .+2             ;.ASCIZ "?"<15><12>"= "
(1) 014230  000750                      BR      1$              ;ZAP THE BUFFER AND LOOP
(1) 014232  111337  015104      3$:     MOVB    (3),.TYPE       ;SET UP FOR TYPING
```

```
(1)  014236  000004  015104                TYPE    ,.TYPE          ;ECHO IT
(1)  014242  122723  000015                CMPB    #15,(3)+        ;CHECK FOR RETURN
(1)  014246  001343                        BNE     2$              ;LOOP IF NOT RETURN
(1)  014250  005737  014274                TST     INHRE
(1)  014254  001401                        BEQ     5$
(1)  014256  000402                        BR      6$
(1)  014260  105063  177777        5$:     CLRB    -1(3)           ;ZAP RETURN (THE 15)
(1)  014264  000004  000012        6$:     TYPE    ,12             ;TYPE A LINE FEED
(1)  014270  012603                        MOV     (6)+,R3         ;RESTORE R3
(1)  014272  000207                        RTS     PC              ;RETURN
(1)
(1)  014274  000000                INHRE:  0
(1)  014276  000020                INPUT:  .BLKW   20              ;TTY INPUT AREA
3693                               ;       $OCTAL          OCTAL TYPEOUT ROUTINE
(1)
(1)                                ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
(1)                                ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
(1)                                ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.
(1)
(1)  014336  012737  170101  014504  BITYP$: MOV     #170101,.PR     ;SET BIT FLAG ANS 16. CHARACTER COUNT
(1)  014344  000411                        BR      .PTIT           ;NOW TYPE IT IN BIT FORM
(1)  014346  112737  000001  014504  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
(1)  014354  000402                        BR      .+6             ;SKIP
(1)  014356  005037  014504        PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
(1)  014362  112737  177772  014505        MOVB    #-6,.PR+1       ;SET COUNT
(1)  014370  010446                .PTIT:  MOV     R4,-(6)         ;SAVE R4
(1)  014372  012704  014506                MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
(1)  014376  105014                        CLRB    (4)             ;CLEAR FIRST BYTE
(1)  014400  000411                        BR      .PRF            ;ROTATE FIRST BIT
(1)  014402  105014                .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
(1)  014404  032737  000100  014504        BIT     #100,.PR        ;BIT TYPING MODE?
(1)  014412  001004                        BNE     .PRF            ;YES - SKIP 2 ROTATES
(1)  014414  006105                        ROL     TTY             ;ROTATE BIT INTO C
(1)  014416  106114                        ROLB    (4)             ;PACK IT
(1)  014420  006105                        ROL     TTY             ;ROTATE BIT INTO C
(1)  014422  106114                        ROLB    (4)             ;PACK IT
(1)  014424  006105                .PRF:   ROL     TTY             ;ROTATE BIT INTO C
(1)  014426  106114                        ROLB    (4)             ;PACK IT
(1)  014430  105714                        TSTB    (4)             ;IS IT ZERO?
(1)  014432  001402                        BEQ     .+6             ;SKIP INC
(1)  014434  105237  014504                INCB    .PR             ;SET FILL SWITCH
(1)  014440  105737  014504                TSTB    .PR             ;CHECK FILL SWITCH
(1)  014444  001402                        BEQ     .+6             ;SKIP BITSET
(1)  014446  152724  000060                BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
(1)  014452  105237  014505                INCB    .PR+1           ;INC COUNT
(1)  014456  001351                        BNE     .PRL            ;REPEAT
(1)  014460  022704  014506                CMP     #.PR+2,R4       ;EMPTY BUFFER?
(1)  014464  001002                        BNE     .+6             ;SKIP IF NOT
(1)  014466  112724  000060                MOVB    #'0,(4)+        ;LOAD 1 ZERO
(1)  014472  105014                        CLRB    (4)             ;NULL TERMINATOR
(1)  014474  000004  014506                TYPE    ,.PR+2          ;TYPE IT
(1)  014500  012604                        MOV     (6)+,R4         ;RESTORE R4
(1)  014502  000207                        RTS     PC              ;RETURN
(1)  014504  000012                .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
 3695  014530  012777  014656  000126  PDOWN$: MOV     #ILLUP,@PUVEC$  ;SET FOR FAST UP
  (1)  014536  012777  000340  000122          MOV     #340,@PUVEC$+2  ;PRIO:7
  (3)  014544  010046                          MOV     R0,-(6)         ;PUSH R0 ON STACK
  (3)  014546  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
  (3)  014550  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
  (3)  014552  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
  (3)  014554  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
  (3)  014556  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
  (1)  014560  010637  014662                  MOV     SP,.SAVR6       ;SAVE SP
  (1)  014564  012777  014574  000072          MOV     #PUP$,@PUVEC$   ;SET UP VECTOR
  (1)  014572  000000                          HALT                    ;WAIT FOR PF
  (1)
  (1)  014574  013706  014662          PUP$:   MOV     .SAVR6,SP       ;GET SP
  (1)  014600  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
  (1)  014602  005201                  1$:     INC     R1              ;WAIT FOR THE INC
  (1)  014604  001376                          BNE     1$              ;OF  WORD
  (3)  014606  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
  (3)  014610  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
  (3)  014612  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
  (3)  014614  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
  (3)  014616  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
  (3)  014620  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
  (1)  014622  012737  014530  000024          MOV     #PDOWN$,@#24    ;SET UP THE POWER DOWN VECTOR
  (1)  014630  012737  000340  000026          MOV     #340,@#26       ;PRIO:7
  (2)  014636  000004  014642                  TYPE    .+2             ;.ASCIZ <15><12>"POWER"
  (1)  014652  000137  006310                  JMP     RESTAR          ;JMP TO USER ADDRESS
  (1)
  (1)  014656  000000                  ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
  (1)  014660  000776                          BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
  (1)
  (1)  014662  000000                  .SAVR6: 0                       ;PUT THE SP HERE
  (1)  014664  000024  000026          PUVEC$: 24,26                   ;POWER UP VECTOR
 3696
 3697
 3698  014670  000002                  YESRT:  RTI                     ;RETURN FROM TRACE TRAP
```

```
3703
3704                                    ;*************************************************************************
3705                                    ;IOT HANDLER - REENTERENT ROUTINE TO INDICATE A FALSE
3706                                    ;              INTERRUPT OR TRAP, OR TO TYPE A MESSAGE
3707                                    ;*************************************************************************
3708
3709  014672  022716  001000           IOTRAP: CMP     #1000,  (SP)    ;CHECK RETURN ADDRESS
3710  014676  002440                          BLT     IOT$            ;BRANCH IF TYPE COMMAND
3711  014700  162716  000004                  SUB     #4,     (SP)    ;GET VECTOR ADDRESS
3712  014704  000004  014710                  TYPE,   .+2             ;TYPE MESSAGE
3713  014710  005015  047125  054105          .ASCIZ  <15><12>"UNEXPECTED INTERUPT TO "
      014716  042520  052103  042105
      014724  044440  052116  051105
      014732  050125  020124  047524
      014740  000040
3714  014742  012605                          MOV     (SP)+,TTY       ;TYPE (SP)+ IN OCTAL
 (1)  014744  004737  014356                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
3715  014750  005726                          TST     (SP)+           ;POP STACK
3716  014752  000004  014756                  TYPE,   .+2             ;TYPE MESSAGE
3717  014756  043040  047522  020115          .ASCIZ  " FROM "
      014764     000
3718             014766                       .EVEN
3719  014766  011605                          MOV     (SP),TTY        ;TYPE (SP) IN OCTAL
 (1)  014770  004737  014356                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
3720  014774  000000                          HALT                    ;FATAL ERROR
3721  014776  000002                          RTI                     ;CONTINUE IF DESIRED
3722
3723                                    ;       $TYPE           MESSAGE TYPEOUT ROUTINE
 (1)
 (1)                                    ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
 (1)                                    ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
 (1)                                    ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
 (1)                                    ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
 (1)                                    ;THE MESSAGE WHICH IS INLINE ASCII.
 (1)
 (1)  015000  010546                    IOT$:   MOV     TTY,-(6)        ;SAVE TTY
 (1)  015002  017605  000002                    MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
 (1)  015006  032705  177400                    BIT     #177400,TTY     ;IS IT A TYPEM?
 (1)  015012  001004                            BNE     1$              ;NO
 (1)  015014  010537  015104                    MOV     TTY,.TYPE       ;GET THE CHARACTER
 (1)  015020  012705  015104                    MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
 (1)  015024  105715                    1$:     TSTB    (TTY)           ;TERMINATOR?
 (1)  015026  001406                            BEQ     2$              ;GET OUT IF SO
 (1)  015030  112537  177566                    MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
 (1)  015034  105737  177564                    TSTB    @#177564        ;IS THE PRINTER READY
 (1)  015040  100375                            BPL     .-4             ;WAIT UNTIL IT IS
 (1)  015042  000770                            BR      1$              ;GET THE NEXT CHARACTER
 (1)  015044  017646  000002            2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
 (1)  015050  062766  000002  000004            ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
 (1)  015056  022666  000002                    CMP     (6)+,2(6)       ;IS IT .+2?
 (1)  015062  001006                            BNE     3$              ;NO
 (1)  015064  062705  000002                    ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
 (1)  015070  042705  000001                    BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
 (1)  015074  010566  000002                    MOV     TTY,2(6)        ;RESTORE ADDRESS
 (1)  015100  012605                    3$:     MOV     (6)+,TTY        ;RESTORE TTY
 (1)  015102  000002                            RTI                     ;RETURN
```

```
                                .TYPE:  0                       ;CHARACTER TYPE LOCATION
3724
(1) 015106 022737 000176 001210 CNTLU:  CMP     #SWREG,SWR
(1) 015114 001023               BNE     1$
(1) 015116 000004 015200        TYPE    ,SWREQ
(2) 015122 013705 000176        MOV     SWREG,TTY       ;TYPE SWREG IN OCTAL
(2) 015126 004737 014346        JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
(1) 015132 000004 015170        TYPE    ,NEWIS
(1) 015136 004537 014002        JSR     R5,READIN
(1) 015142 015336               .WORD   TMP1
(1) 015144 001360               BNE     CNTLU
(1) 015146 022737 000020 015166 CMP     #20,CNT
(1) 015154 001403               BEQ     1$
(1) 015156 013777 015336 164024 MOV     TMP1,@SWR
(1) 015164 000207               1$:     RTS     PC
(1)
(1) 015166 000000               CNT:    0
(1)
(1) 015170 020040 042516 036527 NEWIS:  .ASCIZ  "  NEW= "
(1) 015176 000040
(1) 015200 005015 053523 036522 SWREQ:  .ASCIZ  <15><12>"SWR= "
(1) 015206 000040
(1)
3725
(1) 015210 013746 000006        SUSWRR: MOV     6,-(SP)
(1) 015214 013746 000004                MOV     4,-(SP)
(1) 015220 012737 015240 000004         MOV     #1$,4
(1) 015226 022777 177777 163754         CMP     #-1,@SWR
(1) 015234 001402                       BEQ     2$
(1) 015236 000407                       BR      3$
(1) 015240 022626               1$:     CMP     (SP)+,(SP)+
(1) 015242 012737 000176 001210 2$:     MOV     #SWREG,SWR
(1) 015250 012737 000174 001212         MOV     #DISPREG,DISPLAY
(1) 015256 012637 000004        3$:     MOV     (SP)+,4
(1) 015262 012637 000006                MOV     (SP)+,6
(1) 015266 000207                       RTS     PC
(1)
3726
(1) 015270 022737 000176 001210 KBDINT: CMP     #SWREG,SWR
(1) 015276 001016               BNE     1$
(1) 015300 005037 015336        CLR     TMP1
(1) 015304 113737 177562 015336 MOVB    177562,TMP1
(1) 015312 042737 000200 015336 BICB    #200,TMP1
(1) 015320 122737 000007 015336 CMPB    #7,TMP1
(1) 015326 001003               BNE     1$
(1) 015330 004737 015106        JSR     PC,CNTLU
(1) 015334 000207               1$:     RTS     PC
(1)
(1) 015336 000000               TMP1:   0
(1)
3727
3728 015342 005015 047505 000120 MEOP:   .ASCIZ  <15><12>"EOP"
3729 015346 020043 000000        MNUM:   .ASCIZ  "# "
3730 015351 000040 042523 042514 MSGSEL: .ASCIZ  " SELECT LINE = "
     015356 052103 046040 047111
     015364 020105 020075 000000
```

```
 3731  015371     015  177412  177777  RETURN: .ASCIZ  <15><12><377><377><377>
       015376     000
 3732  015377     040  000040          SPACE:  .ASCIZ  "  "
 3733  015402  005015  046777  044501  MSGMDN: .ASCIZ  <15><12><377>"MAINDEC-11-DZDJB-E   DJ11 EXERCISER"
       015410  042116  041505  030455
       015416  026461  055104  045104
       015424  026502  020105  020040
       015432  045104  030461  042440
       015440  042530  041522  051511
       015446  051105     000
 3734  015451     015  043012  051111  MSGADR: .ASCIZ  <15><12>"FIRST DJ11 ADDRESS:  "
       015456  052123  042040  030512
       015464  020061  042101  051104
       015472  051505  035123  020040
       015500     000
 3735  015501     015  053012  041505  MSGVEC: .ASCIZ  <15><12>"VECTOR ADDRESS:  "
       015506  047524  020122  042101
       015514  051104  051505  035123
       015522  020040     000
 3736  015525     015  047012  027117  MSGNUM: .ASCIZ  <15><12>"NO. OF DJ11'S:  "
       015532  047440  020106  045104
       015540  030461  051447  020072
       015546  000040
 3737  015550  005015  051120  043517  MSGPRG: .ASCIZ  <15><12>"PROGRAM #:  "
       015556  040522  020115  035043
       015564  020040     000
 3738  015567     015  047012  020117  MSGD1:  .ASCIZ  <15><12>"NO DJ11'S!"
       015574  045104  030461  051447
       015602  000041
 3739  015604  005015  051120  043517  MSGP2:  .ASCIZ  <15><12>"PROG2:  CONTINUOUS ECHO EXERCISER"<15><12>
       015612  035062  020040  047503
       015620  052116  047111  047525
       015626  051525  042440  044103
       015634  020117  054105  051105
       015642  044503  042523  006522
       015650  000012
 3740  015652  005015  042452  044103  MSGP3:  .ASCIZ  <15><12>"*ECHO TEST*"<15><12>
       015660  020117  042524  052123
       015666  006452  000012
 3741  015672  026440  000040          MSGDAS: .ASCIZ  " - "
 3742  015676  005015  044523  047514  MALARM: .ASCIZ <15><12>"SILO ALARM LEVEL FOR CSR"<15><12>
       015704  040440  040514  046522
       015712  046040  053105  046105
       015720  043040  051117  041440
       015726  051123  005015     000
 3743          015734                  .EVEN
 3744  015734  000000                  END:    0
 3745          000001                  .END
```

```
ADRNIT  010206          2954    2955    3171#
ALL     006270          2842    2848    2864#
ALMCK   007006          2942    3009#   3025
ALMFLG  001300          2647#   2690*   2940    2971*
BEGIN   005312          2625    2672#
BELL  = 000007          2620#   3638    3641
BINARY  010252          3171    3187#
BITYPS  014336          3693#
BIT0  = 000001          2620#   3026
BIT1  = 000002          2620#
BIT10 = 002000          2620#
BIT11 = 004000          2620#
BIT12 = 010000          2620#
BIT13 = 020000          2620#   2765
BIT14 = 040000          2620#
BIT15 = 100000          2620#
BIT2  = 000004          2620#   3010    3046
BIT3  = 000010          2620#   3027
BIT4  = 000020          2620#   2791    2938    2964    3047
BIT5  = 000040          2620#
BIT6  = 000100          2620#
BIT7  = 000200          2620#
BIT8  = 000400          2620#   2847
BIT9  = 001000          2620#   2961
BUFEND  001264          2641#   3427*   3428*   3547    3619
BUFSIZ  001262          2640#   3428    3466
CNT     015166          3664*   3682*   3724#
CNTLU   015106          2786    3724#   3726
CNTNIT  010210          2691*   2692*   2693*   2956    2957    3035*   3036*   3041*   3043*   3044*   3045*   3172#
CNTTAB= 004313          2666#   3328    3332*   3399*
COUNT   001306          2650#   2998    3012*   3019*   3028    3030    3034    3037    3040    3056    3066
DEVADR  001270          2643#   2703    2730    2770    2788    2807    2918    3243    3438    3501
DISPLA  001212          2633#   3725*
DISPRE  000174          2625#   3725
DJMAP   005452          2703#
DONE    013522          2719    3008    3638#
EMT$    013610          2679    3641#
END     015734          3177    3179    3181    3183    3185    3427    3429    3549    3621    3623    3744#
ERRORS  001202          2631#   2681*   3641*
ERRORS$ 013742          3641    3643#
FORGND  006710          2973    2989#
FORP2   011200          3288#   3294
FORP3   012460          3493#
GETLEN  005754          2766    2784#
GETVEC  005534          2716    2728#
HLT   = 104000          2620#   3068    3136    3153    3367    3382    3580    3594    3599    3605
HLTADR  013740          3641#*
HLTCT$  013736          3641#*  3644*
ICNT    001200          2630#
ILLUP   014656          3695#
INHRE   014274          3659*   3661*   3692#
INPUT   014276          3665    3692#
IOTRAP  014672          2675    3709#
IOT$    015000          3710    3723#
KBDINT  015270          3049    3638    3641    3726#
LEVEL7= 000340          2621#
```

```
M3LARM  015676          3052    3742#
MARK    001260          2639#   2946*   2952    2959*   3266*   3268    3273*   3341*   3343*   3346    3390*   3392*   3395
                        3459*   3462    3477*
M3SK    004312          2665#   2666    2808    3149    3380    3592
MEMX3   012530          3508#   3514
MEOP    015340          3638    3728#
MNUM    015346          2852    3729#
MSGADR  015451          2769    3734#
MSGDAS  015672          3055    3741#
MSGMDN  015402          2767    3733#
MSGNUM  015525          2773    3736#
MSGPRG  015550          2843    3737#
MSGP2   015604          3270    3739#
MSGP3   015652          3467    3740#
MSGSEL  015351          2855    3730#
MSGVEC  015501          2771    3735#
MSG01   015567          2717    3738#
N       =  000040       2563#   3189#   3199#   3201#   3205#
NEWIS   015170          3724#
OPEN    =  000000       2622#   2638
PCNT    001204          2632#   2682*   2683*   3638*
PDOWN$  014530          2677    3695#
PHRASE  010652          3173    3207#
PRINTR  014346          2770    3054    3057    3641    3648    3693#   3724
PRINTS  014356          2772    2774    2854    3693#   3714    3719
PROGAD  006332          2876    2879#
PROGNO  006332          2845    2871    2878#
PROG1   006342          2880    2897#   3006
PROG2   010762          2881    3222#
PROG3   012166          2882    3414#   3613
PS      =  177776       2620#   2899*   2972*   3224*   3279*   3416*   3484*   3685    3687*
PSTEMP  014146          3685*   3687    3690#
PUP$    014574          3695#
PUVEC$  014664          3695#*
P1INIT  006410          2917#
P2INIT  011030          3242#
P2RISR  012012          3308    3361#
P2XISR  011620          3308    3317#
P3INIT  012254          3437#
P3RISR  013300          3528    3574#
P3XISR  013150          3528    3537#
RCVCNT=  003313         2663#   2957*   2996    3157*
RCVISR  010070          3086    3130#
RCVLVL  001274          2645#   2922    3247    3442
RCVTAB  002312          2659#   2955*   3148    3159*   3334    3337    3385    3387*   3398*   3465*   3597    3603    3617*
                        3618*   3619    3621*   3623*   3624    3625*   3626*
READIN  014002          2844    2856    3659#   3724
READ$   014150          3660    3692#
REPORT  005670          2754    2765#
RESTAR  006310          2627    2862    2871#   3638    3695
RETURN  015371          2768    2851    3348    3400    3731#
RISR0   007354          2920    3086#
RISR1   007374          3086#
RISR10  007554          3086#
RISR11  007574          3086#
RISR12  007614          3086#
```

```
RISR13  007634           3086#
RISR14  007654           3086#
RISR15  007674           3086#
RISR16  007714           3086#
RISR17  007734           3086#
RISR2   007414           3086#
RISR3   007434           3086#
RISR4   007454           3086#
RISR5   007474           3086#
RISR6   007514           3086#
RISR7   007534           3086#
R2SR0   011220           3245    3308#
R2SR1   011240           3308#
R2SR10  011420           3308#
R2SR11  011440           3308#
R2SR12  011460           3308#
R2SR13  011500           3308#
R2SR14  011520           3308#
R2SR15  011540           3308#
R2SR16  011560           3308#
R2SR17  011600           3308#
R2SR2   011260           3308#
R2SR3   011300           3308#
R2SR4   011320           3308#
R2SR5   011340           3308#
R2SR6   011360           3308#
R2SR7   011400           3308#
R3SR0   012550           3440    3528#
R3SR1   012570           3528#
R3SR10  012750           3528#
R3SR11  012770           3528#
R3SR12  013010           3528#
R3SR13  013030           3528#
R3SR14  013050           3528#
R3SR15  013070           3528#
R3SR16  013110           3528#
R3SR17  013130           3528#
R3SR2   012610           3528#
R3SR3   012630           3528#
R3SR4   012650           3528#
R3SR5   012670           3528#
R3SR6   012710           3528#
R3SR7   012730           3528#
SAVIT   001214           2634#   2857    2858
SCOPE =  104400          2620#
SELPRO  006162           2841#   2846    2872    2874
SIXBIT  010244           3175    3186#
SPACE   015377           3649    3732#
STACK =  001200          2623#   2672    2898    3223    3415
SUM     001310           2651#   3013*   3030*   3031    3034*   3035    3036    3037*   3038    3040*   3041
SUSWRR  015210           2673    3725#
SVSW0   001220           2638#   2850    2865    2944    3264    3457
SVSW1   001222           2638#
SVSW10  001240           2638#
SVSW11  001242           2638#
SVSW12  001244           2638#
```

```
SVSW13  001246          2638#
SVSW14  001250          2638#
SVSW15  001252          2638#
SVSW16  001254          2638#
SVSW17  001256          2638#
SVSW2   001224          2638#
SVSW3   001226          2638#
SVSW4   001230          2638#
SVSW5   001232          2638#
SVSW6   001234          2638#
SVSW7   001236          2638#
SWITCH  006222          2851#   2861
SWR     001210          2633#   2765    2784    2847    2947    2961    3050    3638    3641    3724*   3725*   3726
SWREG   000176          2625#   2784    3724    3725    3726
SWREQ   015200          3724#
SW10  = 002000          2571#   3638    3641
SW11  = 004000          2571#
SW12  = 010000          2571#   3050
SW13  = 020000          2571#   3641
SW14  = 040000          2571#
SW15  = 100000          2571#
SW8   = 000400          2571#
SW9   = 001000          2571#   3641
TIME    007320          3022    3062#
TIMERA  001302          2648#   3014*   3062*
TIMERB  001304          2649#   3015*   3064*
TIMES   001216          2635#   2694*   3004*   3007*
TMP1    015336          3724    3726#*
TYPE  = 000004          2620#   2717    2767    2768    2769    2771    2773    2843    2851    2852    2855    3052    3055
                        3638    3641    3649    3692    3693    3695    3712    3716    3724
UNITS   001266          2642#   2715*   2746    2774    2787    2806    2860    2864    2969    3277    3482    3500
VECADR  001272          2644#   2740*   2772    2919    3244    3439
XISR0   007364          3086#
XISR1   007404          3086#
XISR10  007564          3086#
XISR11  007604          3086#
XISR12  007624          3086#
XISR13  007644          3086#
XISR14  007664          3086#
XISR15  007704          3086#
XISR16  007724          3086#
XISR17  007744          3086#
XISR2   007424          3086#
XISR3   007444          3086#
XISR4   007464          3086#
XISR5   007504          3086#
XISR6   007524          3086#
XISR7   007544          3086#
XMTCNT  003312          2662#   2663    2956*   2989    3102    3105*   3271*   3324    3327*   3336*   3349*   3401*
XMTISR  007754          3086    3095#
XMTLVL  001276          2646#   2928    3253    3448
XMTTAB  001312          2657#   2900    2954*   3104    3106*   3109*   3225    3270*   3326    3330*   3334*   3335*   3348*
                        3400*   3417    3464*   3493    3544    3545*   3546*   3547    3549*   3550    3553*   3615    3624*
X2SR0   011230          3308#
X2SR1   011250          3308#
X2SR10  011430          3308#
```

```
X2SR11  011450          3308#
X2SR12  011470          3308#
X2SR13  011510          3308#
X2SR14  011530          3308#
X2SR15  011550          3308#
X2SR16  011570          3308#
X2SR17  011610          3308#
X2SR2   011270          3308#
X2SR3   011310          3308#
X2SR4   011330          3308#
X2SR5   011350          3308#
X2SR6   011370          3308#
X2SR7   011410          3308#
X3SR0   012560          3528#
X3SR1   012600          3528#
X3SR10  012760          3528#
X3SR11  013000          3528#
X3SR12  013020          3528#
X3SR13  013040          3528#
X3SR14  013060          3528#
X3SR15  013100          3528#
X3SR16  013120          3528#
X3SR17  013140          3528#
X3SR2   012620          3528#
X3SR3   012640          3528#
X3SR4   012660          3528#
X3SR5   012700          3528#
X3SR6   012720          3528#
X3SR7   012740          3528#
YESRT   014670          3698#
$ENDAD= 013574          2625    3638#
.       = 015736        2625#   2626#   2629#   2657#   2659#   2662#   2665#   3209#   3638    3641#   3692#   3693#   3695
                        3712    3716    3718#   3723    3743#
.BIT    = 132000        2562#   2571    3638    3641
.PR     014504          3693#*
.PRF    014424          3693#
.PRL    014402          3693#
.PTIT   014370          3693#
.SAVR6  014662          3695#*
.TYPE   015104          3692*   3723#*
```

```
BCP      3191#  3199   3205
BITYPE    525#  2620#
BLKBLK   1196#
BUFHDR   1189#
DUMP      515#  2620#  2770   3641   3724
DUMP18    538#
FILBLK   1181#
LNKBLK   1173#
OOT11    1224#
OOT11X   1740#
POP       571#  2620#  3120   3161   3351   3403   3479   3564   3636   3686   3695
PRINT     530#  2620#  3641   3692   3695
PUSH      564#  2620#  3095   3130   3317   3361   3461   3537   3574   3662   3695
SCOPE.    353#
SCOP.     382#
SDUMP     520#  2620#  2772   2774   3714   3719
TRACE    1154#
TRNBLK   1202#
TYPEM     543#
WRITE    1210#
$CATCH    578#
$CMTAG    293#
$CNTL    2475#  3724
$END      606#  3638
$EQUAT    451#  2620
$HLT      725#  3641
$IOCAT   2509#  2625
$IOT     2538#
$KBIN    2459#  3726
$KRAT     134#
$LOADR    397#
$OCTAL    789#  3693
$POWER    871#  3695
$RAND    1074#
$RAND4   1117#
$READ    1023#  3692
$SCOPE    654#
$SETUP    311#
$SRAT     191#
$SWDOC    103#  2571
$SWROF   2497#  2633
$SWRRR   2442#  3725
$TRAP     978#
$TYPE     910#  3723
$URAT     242#
.SCOP     374#
.SCOPE    345#


. ABS.  015736     000

 ERRORS DETECTED:  0

 DZDJBE,DZDJBE/CRF=DZDJBE.MAC,DZDJBE.P11
 RUN-TIME:  5  7  .5 SECONDS
```

RUN-TIME RATIO: 228/14=16.0
CORE USED: 32K (63 PAGES)

D05