DJ11

EXERCISER

MD-11-DZDJB-D

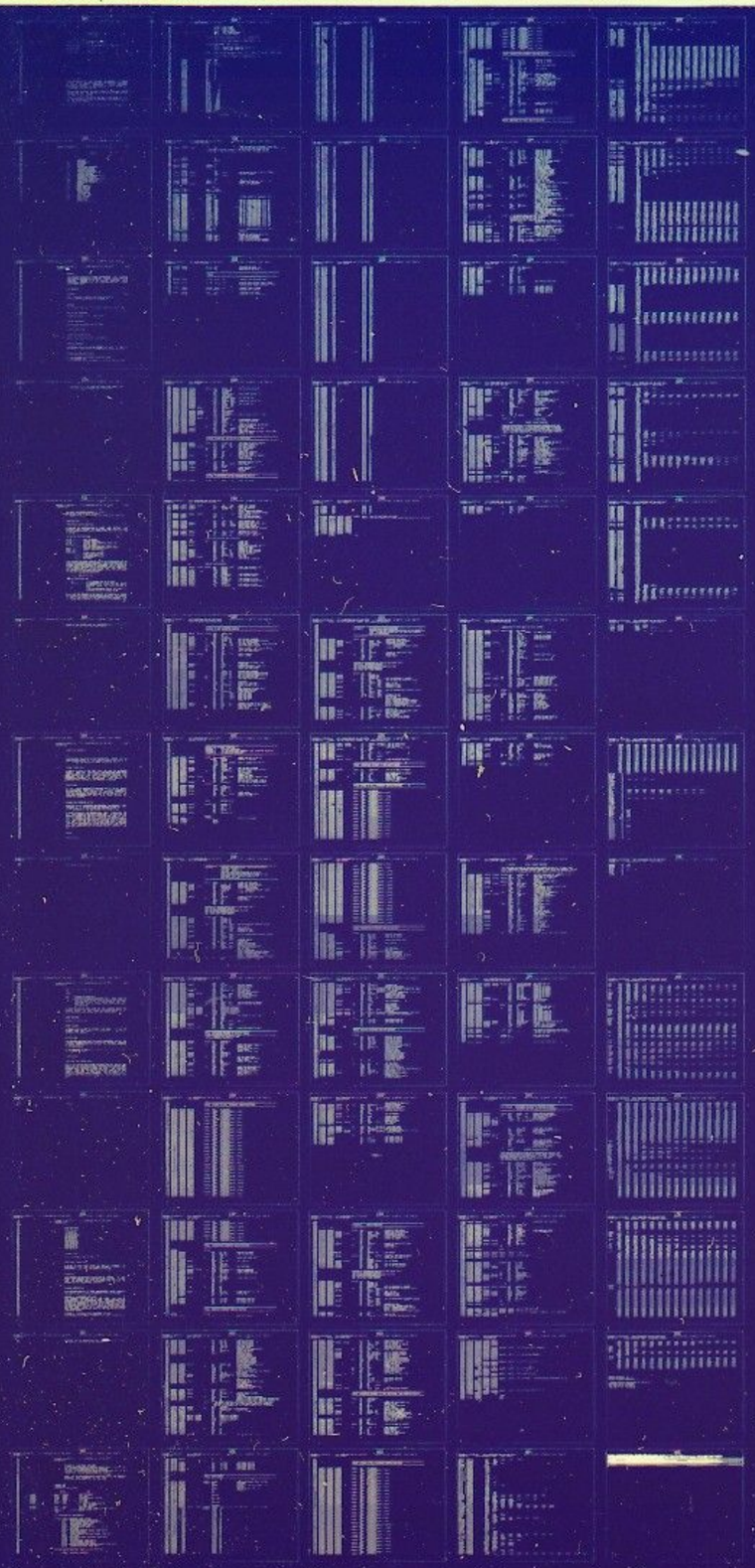
EP DZDJB-D-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDJB-C
PRODUCT NAME: DJ11 EXERCISER AND ON-LINE TESTS
PROGRAM DATE: MAY 1976
MAINTAINER: DIAGNOSTIC GROUP

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.

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

```
137                                   (SEE SEC.  5.1), ALL DOWN FOR WORST CASE, PRESS START.
138                              4) IF SWITCH-LESS PROCESSOR SIMPLY PRESS START.
```

F01

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.

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 UPON COMPLETION OF A PASS OF THE ENTIRE PROGRAM, AND ALTERNATE PASS WILL RUN WITH THE T-BIT SET.

THE SWITCH SETTINGS ARE:

```
SW<15> = 1 ..... HALT ON ERROR
SW<13> = 1 ..... INHIBIT PRINTOUT
SW<12> = 1 ..... INHIBIT TRACE TRAPPING (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
PROG1 ONLY:
SW<2:0>= 0 ..... BINARY COUNT PATTERN
         1 ..... "THE QUICK BROWN 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 TRTRAP (PROG1 ONLY)

IF SW<12> IS ON A 0, THE T-BIT WILL BE SET ON ALTERNATE PASSES. WHEN THE T-BIT IS SET, THE PROCESSOR TRAPS AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTI" OR "RTT" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTIONS. THIS SEQUENCE IS CONTINUED UNTIL THE END OF THE PROGRAM IS REACHED.

5.2.3 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 HALT, DISPLAYING THE DJ11 UNIT NUMBER IN THE DATA LIGHTS (R0). AT THIS TIME THE OPERATOR SELECTS VIA THE SWITCH REGISTER THE LINES OF THAT DJ11 HE WISHES TO RUN (I.E. SW<0> ON A 1 SELECTS LINE 0, ETC.). THE OPERATOR THEN PRESSES "CONTINUE" AND THE PROGRAM WILL HALT FOR THE NEXT DJ11, IF ANY MORE EXIST. WHEN ALL LINES OF THE EXISTING DJ11'S HAVE BEEN SELECTED, THE PROGRAM WILL HALT ONCE MORE WITH 177777 IN THE DATA LIGHTS (R0) TO ALLOW THE OPERATOR TO SET THE DYNAMIC SWITCH SETTINGS (SEE SEC. 5.1).

6. ERRORS

6.1 ERROR PRINTOUT

```
253
254                                     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" (LOC 1202). 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 (DH11'S WILL BE FIRST, STARTING AT 160010, THEN THE DJ11'S) AND THAT THE VECTOR ADDRESSES ALL BE CONTIGUOUS.

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 8 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 WITHOUT TRACE TRAPPING. FOR MULTIPLE DJ11'S, MULTIPLY THESE TIMES BY THE NUMBER OF UNITS SELECTED FOR TEST.

```
311
312                                                    APPROX
```

BAUD	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 "PASSES" (LOC 1204,1206). 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 8 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.

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.
```

```
                        DESCRIPTION

                                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.
                              %

                        .ENABLE ABS
                        .ENABLE AMA
                        ;PROGRAM BY KEN CHAPMAN

                        ;     SWITCH                    USE
                        ;     ------                    ---------------------
100000                  SW15=   100000          ;HALT ON ERROR
040000                  SW14=   40000           ;NOT USED
020000                  SW13=   20000           ;INHIBIT ERROR TYPEOUTS
010000                  SW12=   10000           ;NOT USED
004000                  SW11=   4000            ;NOT USED
002000                  SW10=   2000            ;0 - BELL ON PASS COMPLETE
                                                ;1 - BELL ON ERROR
001000                  SW9=    1000            ;ON-LINE (PROG1)
000400                  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
```

```
                                        BIT9-11 LINE NUMBER
                                        BIT12   PARITY ERROR
                                        BIT13   FRAMING ERROR
                                        BIT14   UART OVERRUN ERROR
                                        BIT15   CHARACTER PRESENT

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

                                        BIT0-15 STOP THE SCANNER ON CORRESPONDING LINE

                                TRANSMITTER BUFFER (TBUF) XXXXX6

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

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

                                        BIT0-15 TRNSMIT A BREAK ON CORRESPONDING LINE!

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

```
                    000000                  . =     0                               ;TRAP CATCHER IN LOCATIONS 0 THRU 776
000000  000000  000000                      0,0                                     ;LOCATIONS 0 AND 2 CONTAIN "HALT" INSTRUCTIONS
                                            ;LOCATIONS 4 THRU 56 CONTAIN ".+2" AND "HALT" IN EVERY VECTOR
                                            ;LOCATIONS 60 THRU 776 CONTAIN ".+2" AND "IOT" IN EVERY VECTOR

                    000046                  . =     46
000046  013310                              $ENDAD

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

                    000200                  . =     200
000200  000137  005306                      JMP     BEGIN                           ;200 ALWAYS IS THE STARTING ADDRESS
                    001000                  . =     1000                            ;RESTART ADDRESS
001000  000137  006322                      JMP     RESTAR

                    001200                  . =     1200
001200  000000                             ICNT:   0                               ;ITERATION COUNT-LH, TEST NO.-RH
001202  000000                             ERRORS: 0                               ;ERROR COUNT REGISTER
001204  000000  000000                     PCNT:   0,0                             ;PASS COUNT REGISTER


001210  177570                             SWR:    177570
001212  177570                             DISPLAY:177570

001214  000000                             SAVIT:  0
001216  000020                             TIMES:  20                              ;MINIMUM NUMBER OF MESSAGES (PROG1)
001220  000000                             SWRSAV: 0
001222  000000                             SVSW0:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #0
001224  000000                             SVSW1:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #1
001226  000000                             SVSW2:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #2
001230  000000                             SVSW3:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #3
001232  000000                             SVSW4:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #4
001234  000000                             SVSW5:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #5
001236  000000                             SVSW6:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #6
001240  000000                             SVSW7:  OPEN                            ;MAP OF LINES SELECTED, DJ11 #7
001242  000000                             SVSW10: OPEN                            ;MAP OF LINES SELECTED, DJ11 #10
001244  000000                             SVSW11: OPEN                            ;MAP OF LINES SELECTED, DJ11 #11
001246  000000                             SVSW12: OPEN                            ;MAP OF LINES SELECTED, DJ11 #12
001250  000000                             SVSW13: OPEN                            ;MAP OF LINES SELECTED, DJ11 #13
001252  000000                             SVSW14: OPEN                            ;MAP OF LINES SELECTED, DJ11 #14
001254  000000                             SVSW15: OPEN                            ;MAP OF LINES SELECTED, DJ11 #15
001256  000000                             SVSW16: OPEN                            ;MAP OF LINES SELECTED, DJ11 #16
001260  000000                             SVSW17: OPEN                            ;MAP OF LINES SELECTED, DJ11 #17
001262  000000                             MARK:   0
001264  000020                             BUFSIZ: 20                              ;RECEIVE BUFFER SIZE (PROG3)
001266  000017                             BUFMSK: 17                              ;BUFSIZ-1

001270  000001                             UNITS:  1                               ;NUMBER OF UNITS ON THE SYSTEM
001272  160010                             DEVADR: 160010                          ;FIRST DEVICE ADDRESS
001274  000300                             VECADR: 300                             ;FIRST VECTOR ADDRESS
001276  160010                             FLTADR: 160010                          ;FIRST FLOATING DEVICE ADDRESS
001300  000300                             FLTVEC: 300                             ;FIRST FLOATING VECTOR ADDRESS
```

```
                                RCVLVL: 240                     ;RECEIVER BR LEVEL = 5
                                XMTLVL: 240                     ;TRANSMITTER BR LEVEL = 5

                                ;*******************************************************************************
                                ;TABLES
                                ;*******************************************************************************

001306  000400                  XMTTAB: .BLKW   400             ;TRANSMIT DATA POINTER TABLE

002306  000400                  RCVTAB: .BLKW   400             ;RECEIVE DATA POINTER TABLE (PROG1)
                                                                ;RECEIVE DATA TABLE (PROG2 AND PROG3)

003306  000400                  XMTCNT: .BLKW   400             ;TRANSMIT DATA COUNTER
        003307                  RCVCNT=XMTCNT+1                 ;RECEIVE DATA COUNTER

004306  000400                  MASK:   .BLKW   400             ;CHARACTER MASK TABLE
        004307                  CNTTAB=MASK+1                   ;ITERATION COUNT AND FLAGS
```

```
        005306  012706  001200          BEGIN:  MOV     #ICNT,  SP      ;SET UP STACK POINTER
        005312  004737  014726                  JSR     PC,SUSWRR
        005316  012700  000014                  MOV     #14,RO
        005322  012720  014406                  MOV     #YESRT,(RO)+    ;TRACE TRAP VECTOR (14)
        005326  012720  000340                  MOV     #340,(RO)+
        005332  012720  014410                  MOV     #IOTRAP,(RO)+   ;IOT VECTOR (20)
        005336  012720  000340                  MOV     #340,   (RO)+
        005342  012720  014246                  MOV     #PDOWN$,(RO)+   ;POWER FAIL VECTOR (24)
        005346  012720  000340                  MOV     #340,   (RO)+
        005352  012720  013326                  MOV     #EMT$,  (RO)+   ;EMT VECTOR (30)
        005356  012720  000340                  MOV     #340,   (RO)+
        005362  012737  005420  000010          MOV     #1$,@#10
        005370  006700                          SXT     RO
        005372  012737  000006  014406          MOV     #RTT,@#YESRT
        005400  012737  000006  000006          MOV     #RTT,@#6
        005406  012737  000400  177774          MOV     #400,@#177774   ;SET UP STACK LIMIT TO 1000
        005414  005037  000006                  CLR     @#6
        005420  012737  000012  000010  1$:     MOV     #12,@#10
        005426  005037  001202                  CLR     ERRORS          ;CLEAR ERROR COUNTER
        005432  005037  001204                  CLR     PCNT            ;CLEAR PASS COUNTER
        005436  005037  001206                  CLR     PCNT+2
        005442  012700  000300                  MOV     #300,   RO      ;START OF FLOATING VECTOR AREA
        005446  005720                  2$:     TST     (RO)+           ;UPDATE POINTER
        005450  010060  177776                  MOV     RO,     -2(RO)  ;PUT ".+2" IN EACH VECTOR
        005454  012720  000004                  MOV     #IOT,   (RO)+   ;AND "IOT"
        005460  022700  001000                  CMP     #1000,  RO      ;CHECK FOR END OF FLOATING VECTOR AREA
        005464  003370                          BGT     2$              ;BRANCH IF MORE

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

        005466  013700  001276          DJMAP:  MOV     FLTADR, RO      ;GET FIRST FLOATING ADDRESS
        005472  012702  000001                  MOV     #1,     R2      ;COUNTER FOR DJ11'S
        005476  012737  000002  000006          MOV     #RTI,   @#6     ;RTI WHEN TIME-OUT
        005504  005001                  5$:     CLR     R1              ;SET UP COUNTER
        005506  000261                  1$:     SEC                     ;SET CARRY
        005510  005710                          TST     (RO)            ;CHECK FOR A DEVICE
        005512  103404                          BCS     7$              ;BRANCH IF NONE
        005514  062700  000010          6$:     ADD     #10,    RO      ;GO TO NEXT DEVICE ADDRESS
        005520  005201                          INC     R1              ;COUNT DJ11'S
        005522  000771                          BR      1$              ;LOOK FOR MORE

        005524  005037  000006          7$:     CLR     @#6             ;RESTORE TIMEOUT VECTOR
        005530  010137  001270                  MOV     R1,     UNITS   ;SAVE COUNT
        005534  001005                          BNE     GETVEC
        005536  000004  015254                  TYPE,   MSG01           ;TYPE "NO DJ11'S!"
        005542  000000                          HALT                    ;FATAL ERROR
        005544  000137  013242                  JMP     @#DONE          ;RESTART

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

        005550  013746  000020          GETVEC: MOV     @#20,   -(SP)   ;SAVE IOT VECTOR
```

```
                                        MOV     #1$,    @#20    ;RESET IOT VECTOR
                                        MOV     DEVADR, R1      ;FIRST DJ ADDRESS
                                        MOV     #40400, (R1)    ;SET CSR
                                                                ;BIT8= TRANS SCAN ENABLE
                                                                ;BIT14= TRANS INTERRUPT ENABLE
                                        MOV     #1,     4(R1)   ;TCR, LINE 0
                                        WAIT                    ;WAIT FOR AN INTERRUPT
                                        BR      3$              ;CONTINUE AFTER INTERUPT

                                1$:     MOV     (SP),   R2      ;SAVE VECTOR ADR. (+4)
                                        SUB     #10,    (SP)    ;REPOSITION ADR TO RCV. VEC.
                                        MOV     (SP),   VECADR  ;SAVE FIRST VECTOR
                                        CMP     (SP)+,  (SP)+   ;RESET STACK FROM IOT
                                        RTI                     ;RETURN FROM INITIAL INTERUPT

                                3$:     MOV     (SP)+,  @#20    ;RESTORE IOT VECTOR
                                        TST     -(R2)           ;POINT TO XMT. VEC. +2
                                        MOV     UNITS,  R3      ;SET UP UNIT COUNTER

                                ;CHECK THAT VECTORS ARE CONTIGUOUS

                                2$:     CLR     4(R1)           ;CLEAR TCR
                                        CLR     (R1)            ;CLEAR CSR
                                        MOV     #IOT,   (R2)    ;RESTORE IOT TO XMT. VEC.+2
                                        DEC     R3              ;CHECK FOR MORE DJ11'S
                                        BEQ     REPORT          ;BRANCH IF DONE
                                        ADD     #10,    R1      ;UPDATE DJ ADR. POINTER
                                        ADD     #10,    R2      ;UPDATE VECTOR POINTER
                                        MOV     #RTI,   (R2)    ;RTI ON INTERRUPT
                                        MOV     #40400, (R1)    ;SET CSR
                                        MOV     #1,     4(R1)   ;TCR LINE 0
                                        WAIT                    ;WAIT FOR AN INTERRUPT
                                        BR      2$

                                ;REPORT CONFIGURATION

                        REPORT: BIT     #BIT13, @SWR    ;CHECK FOR INHIBIT TYPOUT
                                BNE     GETLEN          ;SKIP REPORT IF SET
                                TYPE,   MSGMDN
                                TYPE,   MSGADR
                                MOV     DEVADR,TTY      ;TYPE DEVADR IN OCTAL
                                JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
                                TYPE,   MSGVEC
                                MOV     VECADR,TTY      ;TYPE VECADR IN OCTAL
                                JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
                                TYPE,   MSGNUM
                                MOV     UNITS,TTY       ;TYPE UNITS IN OCTAL
                                JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
```

```
                                ;*****************************************************************
                                ;ROUTINE TO MAP CHARACTER LENGTHS
                                ;*****************************************************************

005764  022737  000176  001210  GETLEN: CMP     #SWREG,SWR
005772  001002                          BNE     6$
005774  004737  014624                  JSR     PC,CNTLU
006000  013700  001270          6$:     MOV     UNITS, R0       ;SET UP UNIT COUNTER
006004  013701  001272                  MOV     DEVADR, R1      ;SET UP DEVICE ADDRESS POINTER
006010  012702  000001          1$:     MOV     #1,    R2       ;SET UP LINE MARKER
006014  012711  000415                  MOV     #415,  (R1)     ;RCV ENB, CMOS, MAINT., TRANS SCAN ENB
006020  032711  000020          10$:    BIT     #BIT4, (R1)     ;WAIT FOR MOS TO CLEAR
006024  001375                          BNE     10$
006026  010261  000004          2$:     MOV     R2,    4(R1)    ;TRANS CONTROL, ONE LINE AT A TIME
006032  005711                  3$:     TST     (R1)            ;WAIT FOR TRANS READY
006034  100376                          BPL     3$
006036  012761  000377  000006          MOV     #377,  6(R1)    ;SEND A RUBOUT
006044  006302                          ASL     R2              ;SKIP 4 LINES
006046  006302                          ASL     R2
006050  006302                          ASL     R2
006052  006302                          ASL     R2
006054  103364                          BCC     2$              ;BRANCH BACK IF MORE LINES
006056  005061  000004                  CLR     4(R1)           ;CLEAR TCR
006062  062701  000010                  ADD     #10,   R1       ;UPDATE POINTER TO NEXT UNIT
006066  005300                          DEC     R0              ;CHECK FOR MORE UNITS
006070  001347                          BNE     1$
006072  013700  001270                  MOV     UNITS, R0       ;SET UP UNIT COUNTER
006076  013701  001272                  MOV     DEVADR, R1      ;SET UP DEVICE ADDRESS POINTER
006102  012702  004306                  MOV     #MASK, R2       ;SET UP CHAR LEN TABLE POINTER
006106  012703  000004          4$:     MOV     #4,    R3       ;SET UP CHAR COUNTER
006112  016104  000002          5$:     MOV     2(R1), R4       ;SAVE AND CHECK CHAR PRESENT
006116  100375                          BPL     5$
006120  010405                          MOV     R4,    R5       ;DUP DATA
006122  000305                          SWAB    R5              ;LINE # IN LOW BYTE
006124  042705  177760                  BIC     #177760,R5      ;CLEAR ALL BUT LINE #
006130  006305                          ASL     R5              ;*2
006132  060205                          ADD     R2,    R5       ;MAKE POINTER TO CHAR TABLE
006134  105104                          COMB    R4              ;MAKE DATA INTO MASK
006136  042704  174000                  BIC     #174000,R4      ;CLEAR UPPER BYTE
006142  010425                          MOV     R4,    (R5)+    ;SAVE THE MASK
006144  010425                          MOV     R4,    (R5)+    ;SAVE THE MASK
006146  010425                          MOV     R4,    (R5)+    ;SAVE THE MASK
006150  010425                          MOV     R4,    (R5)+    ;SAVE THE MASK
006152  005303                          DEC     R3              ;COUNT TO 4
006154  001356                          BNE     5$
006156  062701  000010                  ADD     #10,   R1       ;ADDRESS POINTER TO NEXT DJ
006162  062702  000040                  ADD     #40,   R2       ;CHAR LEN TABLE POINTER
006166  005300                          DEC     R0              ;COUNT UNITS
006170  001346                          BNE     4$              ;BRANCH BACK IF MORE
```

```
                                        ;*****************************************************************************
                                        ;SELECT THE PROGRAM TO BE RUN
                                        ;PROGRAM 1:     OFF-LINE EXERCISER
                                        ;PROGRAM 2:     ON-LINE EXERCISER (TRANSMIT LAST CHARACTER RECEIVED)
                                        ;PROGRAM 3:     ON-LINE ECHO EXERCISER
                                        ;*****************************************************************************

006172  005737  000042          SELPRO: TST     @#42            ;CHECK FOR ACT 11 OR DDP
006176  001051                          BNE     RESTAR          ;BRANCH IF MONITOR
006200  000004  015235                  TYPE,   MSGPRG
006204  004537  013520                  JSR     R5,     READIN  ;READ A NUMBER FROM THE CTY
006210  006344                          .WORD   PROGNO
006212  001367                          BNE     SELPRO
006214  013700  001270                  MOV     UNITS,  R0      ;SET UP UNIT COUNTER
006220  012701  001222                  MOV     #SVSW0,R1       ;SET UP SWITCH TABLE POINTER
006224  012721  177777          1$:     MOV     #177777,(R1)+   ;SET ALL LINES
006230  005300                          DEC     R0              ;COUNT UNITS
006232  001374                          BNE     1$              ;BRANCH IF MORE
006234  032777  000400  172746          BIT     #BIT8,  @SWR    ;CHECK FOR SW<8>, SELECT LINES
006242  001427                          BEQ     RESTAR          ;BRANCH IF NOT
006244  005000                          CLR     R0              ;SET UP UNIT COUNTER, DISPLAY
006246  012701  001222                  MOV     #SVSW0,R1       ;SET UP SWITCH TABLE POINTER
006252                          SWITCH:
006252  000004  006256                  TYPE    .+2             ;.ASCIZ <15><12>"SELECT LINE = "
006300  004537  013520                  JSR     R5,READIN
006304  001214                          .WORD   SAVIT
006306  013721  001214                  MOV     SAVIT,(R1)+
006312  005200                          INC     R0              ;COUNT UNITS
006314  020037  001270                  CMP     R0,     UNITS   ;CHECK FOR MORE UNITS
006320  001354                          BNE     SWITCH          ;BRANCH IF MORE


                                        .SBTTL          RESTART POINT

006322  013700  006344          RESTAR: MOV     PROGNO, R0
006326  001721                          BEQ     SELPRO
006330  022700  000003                  CMP     #3,     R0
006334  103716                          BLO     SELPRO
006336  006300                          ASL     R0
006340  000170  006344                  JMP     @PROGAD (R0)

006344                          PROGNO:
006344  000001                  PROGAD: 1                       ;DEFAULT TO PROGRAM 1
006346  006354                          PROG1
006350  010462                          PROG2
006352  011666                          PROG3
```

```
                                ;*****************************************************************
                                ;PROGRAM 1:      TRANSMIT AND RECEIVE ALL LINES SIMULTANIOUSLY
                                ;               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 8 TIMES
                                ;               PER PASS.  I.E. THE FAST LINES MAY GET
                                ;               FOR THE SLOW LINES
                                ;*****************************************************************

006354  000005                  PROG1:  RESET                   ;CLEAR OUT THE WORLD
006356  012706  001200                  MOV     #ICNT,  SP      ;RESET THE STACK POINTER
006362  052737  000340  177776          BIS     #340,@#PS       ;PROCESSOR TO LEVEL 7
006370  012701  001306                  MOV     #XMTTAB,R1      ;FIRST TABLE POINTER
006374  012702  001400                  MOV     #1400,R2                ;LENGTH OF TABLES (WORDS)
006400  005021                  1$:     CLR     (R1)+           ;CLEAR THE TABLE
006402  005302                          DEC     R2
006404  001375                          BNE     1$
006406  012702  000400                  MOV     #400,R2         ;LENGTH OF MASK/COUNT TABLE
006412  005201                  2$:     INC     R1              ;SKIP MASK
006414  105021                          CLRB    (R1)+           ;CLEAR COUNT
006416  005302                          DEC     R2
006420  001374                          BNE     2$

                                ;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

006422  005000                  P1INIT: CLR     R0
006424  013701  001272                  MOV     DEVADR, R1
006430  013702  001274                  MOV     VECADR, R2
006434  012703  006756                  MOV     #RISR0, R3
006440  010322                  11$:    MOV     R3,     (R2)+   ;SET UP RECEIVER INTERUPT VECTOR
006442  013722  001302                  MOV     RCVLVL, (R2)+
006446  022323                          CMP     (R3)+,  (R3)+   ;ADD 4 TO R3
006450  010113                          MOV     R1,     (R3)    ;ADDRESS OF CSR
006452  062723  000002                  ADD     #2,     (R3)+   ;ADDRESS OF RBUF
006456  005723                          TST     (R3)+
006460  010322                          MOV     R3,     (R2)+   ;SET UP TRANSMITTER INTERUPT VECTOR
006462  013722  001304                  MOV     XMTLVL, (R2)+
006466  022323                          CMP     (R3)+,  (R3)+
006470  010123                          MOV     R1,     (R3)+   ;ADDRESS OF CSR
006472  005011                          CLR     (R1)            ;CLEAR CSR
006474  032777  001000  172506          BIT     #BIT9,  @SWR    ;CHECK FOR ON-LINE
006502  001002                          BNE     12$             ;BRANCH IF ON-LINE
006504  012711  000004                  MOV     #BIT2,  (R1)    ;SET THE MAINTENANCE BIT
006510  052711  050510          12$:    BIS     #50510, (R1)    ;SET UP CSR
                                                                ;BIT3 = CLEAR MOS
                                                                ;BIT6 = RECEIVER INTERUPT ENABLE
                                                                ;BIT8 = TRANSMITTER SCAN ENABLE
                                                                ;BIT12 = STATUS ENABLE
                                                                ;BIT14 = TRANSMITTER INTERUPT ENABLE
006514  032711  000020          13$:    BIT     #BIT4,  (R1)    ;CHECK FOR MOS TO CLEAR
```

```
808  006520  001375                          BNE     13$
809  006522  052711  000001                  BIS     #1,(R1)          ;SET RCV ENABLE
810  006526  022121                          CMP     (R1)+,  (R1)+    ;UPDATE POINTER
811  006530  006300                          ASL     R0               ;UNIT # * 2
812  006532  016011  001222                  MOV     SVSW0(R0),(R1)   ;SET TCR BITS
813  006536  006200                          ASR     R0               ;RESTORE UNIT COUNTER
814  006540  012737  000001  001262          MOV     #1,     MARK     ;SET UP MARKER
815  006546  017705  172436                  MOV     @SWR,   R5       ;GET SWITCH SETTINGS
816  006552  042705  177770                  BIC     #177770,R5       ;MASK MESSAGE #
817  006556  006305                          ASL     R5
818  006560  006305                          ASL     R5
819  006562  012304                          MOV     (R3)+,  R4       ;SET UP OFFSET TO TABLES
820  006564  033711  001262          14$:    BIT     MARK,   (R1)     ;CHECK FOR LINE SELECTED
821  006570  001417                          BEQ     15$
822  006572  016564  007714  001306          MOV     ADRNIT(5),XMTTAB(4)
823  006600  016564  007714  002306          MOV     ADRNIT(5),RCVTAB(4)
824  006606  116564  007716  003306          MOVB    CNTNIT(5),XMTCNT(4)
825  006614  116564  007716  003307          MOVB    CNTNIT(5),RCVCNT(4)
826  006622  113764  001216  004307          MOVB    TIMES,  CNTTAB(4)
827  006630  005724                  15$:    TST     (R4)+            ;INC OFFSET TO NEXT LINE
828  006632  006337  001262                  ASL     MARK
829  006636  103352                          BCC     14$
830  006640  022121                          CMP     (R1)+,  (R1)+    ;ADD 4
831  006642  005200                          INC     R0
832  006644  020037  001270                  CMP     R0,     UNITS
833  006650  001273                          BNE     11$
834  006652  042737  000140  177776          BIC     #140,   @#PS     ;LOWER PROCESSOR PRIORITY
835
836                                  ;*****************************************************************************
837                                  ;PROG1  BACKGROUND PROGRAM TO MONITOR TABLES
838                                  ;*****************************************************************************
839                                  ; NOTE - PROGRAM MAY HANG IN A LOOP.
840                                  ; IF THIS HAPPENS, RUN DZDJA.
841
842  006660  012701  004307          FORGND: MOV     #CNTTAB,R1
843  006664  012702  000400                  MOV     #400,R2
844  006670  105711                  21$:    TSTB    (R1)             ;CHECK FOR COUNT TABLE CLR
845  006672  001376                          BNE     21$              ;BRANCH IF NOT
846  006674  062701  000002                  ADD     #2,R1            ;GO TO NEXT LINE ENTRY
847  006700  005302                          DEC     R2               ;COUNT LINES
848  006702  001372                          BNE     21$              ;BRANCH IF MORE LINES
849  006704  012701  004307                  MOV     #CNTTAB,R1
850  006710  012702  000400                  MOV     #400,R2
851  006714  112721  000200          22$:    MOVB    #200,(R1)+       ;SET ALL DONE FLAG
852  006720  005201                          INC     R1               ;INC TO NEXT LINE ENTRY
853  006722  005302                          DEC     R2               ;COUNT LINES
854  006724  001373                          BNE     22$              ;BRANCH IF MORE LINES
855  006726  012701  003307                  MOV     #RCVCNT,R1
856  006732  012702  000400                  MOV     #400,R2
857  006736  105711                  23$:    TSTB    (R1)             ;CHECK FOR RECEIVER DONE
858  006740  001376                          BNE     23$              ;BRANCH IF NOT DONE
859  006742  062701  000002                  ADD     #2,R1            ;GO TO NEXT LINE ENTRY
860  006746  005302                          DEC     R2               ;COUNT LINES
861  006750  001372                          BNE     23$              ;BRANCH IF MORE LINES
862  006752  000137  013242                  JMP     @#DONE           ;SKIP ISR'S
863
```

```
                                        ;*****************************************************************************
                                        ;PROG1  LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
                                        ;*****************************************************************************

006756  004037  007472        RISR0:  JSR     R0,RCVISR
006762  160012  000000                .WORD   <160012+<0*10>>,<40*0>
006766  004037  007356        XISR0:  JSR     R0,XMTISR
006772  160010  000000                .WORD   <160010+<0*10>>,<40*0>
006776  004037  007472        RISR1:  JSR     R0,RCVISR
007002  160022  000040                .WORD   <160012+<1*10>>,<40*1>
007006  004037  007356        XISR1:  JSR     R0,XMTISR
007012  160020  000040                .WORD   <160010+<1*10>>,<40*1>
007016  004037  007472        RISR2:  JSR     R0,RCVISR
007022  160032  000100                .WORD   <160012+<2*10>>,<40*2>
007026  004037  007356        XISR2:  JSR     R0,XMTISR
007032  160030  000100                .WORD   <160010+<2*10>>,<40*2>
007036  004037  007472        RISR3:  JSR     R0,RCVISR
007042  160042  000140                .WORD   <160012+<3*10>>,<40*3>
007046  004037  007356        XISR3:  JSR     R0,XMTISR
007052  160040  000140                .WORD   <160010+<3*10>>,<40*3>
007056  004037  007472        RISR4:  JSR     R0,RCVISR
007062  160052  000200                .WORD   <160012+<4*10>>,<40*4>
007066  004037  007356        XISR4:  JSR     R0,XMTISR
007072  160050  000200                .WORD   <160010+<4*10>>,<40*4>
007076  004037  007472        RISR5:  JSR     R0,RCVISR
007102  160062  000240                .WORD   <160012+<5*10>>,<40*5>
007106  004037  007356        XISR5:  JSR     R0,XMTISR
007112  160060  000240                .WORD   <160010+<5*10>>,<40*5>
007116  004037  007472        RISR6:  JSR     R0,RCVISR
007122  160072  000300                .WORD   <160012+<6*10>>,<40*6>
007126  004037  007356        XISR6:  JSR     R0,XMTISR
007132  160070  000300                .WORD   <160010+<6*10>>,<40*6>
007136  004037  007472        RISR7:  JSR     R0,RCVISR
007142  160102  000340                .WORD   <160012+<7*10>>,<40*7>
007146  004037  007356        XISR7:  JSR     R0,XMTISR
007152  160100  000340                .WORD   <160010+<7*10>>,<40*7>
007156  004037  007472        RISR10: JSR     R0,RCVISR
007162  160112  000400                .WORD   <160012+<10*10>>,<40*10>
007166  004037  007356        XISR10: JSR     R0,XMTISR
007172  160110  000400                .WORD   <160010+<10*10>>,<40*10>
007176  004037  007472        RISR11: JSR     R0,RCVISR
007202  160122  000440                .WORD   <160012+<11*10>>,<40*11>
007206  004037  007356        XISR11: JSR     R0,XMTISR
007212  160120  000440                .WORD   <160010+<11*10>>,<40*11>
007216  004037  007472        RISR12: JSR     R0,RCVISR
007222  160132  000500                .WORD   <160012+<12*10>>,<40*12>
007226  004037  007356        XISR12: JSR     R0,XMTISR
007232  160130  000500                .WORD   <160010+<12*10>>,<40*12>
007236  004037  007472        RISR13: JSR     R0,RCVISR
007242  160142  000540                .WORD   <160012+<13*10>>,<40*13>
007246  004037  007356        XISR13: JSR     R0,XMTISR
007252  160140  000540                .WORD   <160010+<13*10>>,<40*13>
007256  004037  007472        RISR14: JSR     R0,RCVISR
007262  160152  000600                .WORD   <160012+<14*10>>,<40*14>
007266  004037  007356        XISR14: JSR     R0,XMTISR
007272  160150  000600                .WORD   <160010+<14*10>>,<40*14>
```

```
                                ;RISR15: JSR     R0,RCVISR
                                ;        .WORD   <160012+<15*10>>,<40*15>
                                ;XISR15: JSR     R0,XMTISR
                                ;        .WORD   <160010+<15*10>>,<40*15>
                                ;RISR16: JSR     R0,RCVISR
                                ;        .WORD   <160012+<16*10>>,<40*16>
                                ;XISR16: JSR     R0,XMTISR
                                ;        .WORD   <160010+<16*10>>,<40*16>
                                ;RISR17: JSR     R0,RCVISR
                                ;        .WORD   <160012+<17*10>>,<40*17>
                                ;XISR17: JSR     R0,XMTISR
                                ;        .WORD   <160010+<17*10>>,<40*17>

                                ;*******************************************************************************
                                ;PROG1  TRANSMITTER INTERRUPT SERVICE ROUTINE
                                ;*******************************************************************************

007356                          XMTISR:
007356  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
007360  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
007362  012001                          MOV     (R0)+,R1
007364  001711                  1$:     TST     (R1)            ;CHECK FOR TRANS READY
007366  100035                          BPL     4$
007370  116102  000007                  MOVB    7(R1),R2        ;GET LINE NO.
007374  006302                          ASL     R2
007376  061002                          ADD     (R0),R2
007400  105762  003306                  TSTB    XMTCNT(2)       ;TST FOR ZERO
007404  001410                          BEQ     2$
007406  117261  001306  000006          MOVB    @XMTTAB(2),6(1) ;SEND A CHARACTER
007414  105362  003306                  DECB    XMTCNT(2)       ;COUNT CHARACTERS
007420  005262  001306                  INC     XMTTAB(2)       ;UPDATE TABLE POINTER
007424  000757                          BR      1$
007426  010346                  2$:     MOV     R3,-(SP)
007430  005062  001306                  CLR     XMTTAB(2)       ;CLEAR TABLE POINTER
007434  161002                          SUB     (R0),R2
007436  006202                          ASR     R2
007440  005003                          CLR     R3
007442  000261                          SEC
007444  006103                  3$:     ROL     R3
007446  005302                          DEC     R2
007450  100375                          BPL     3$
007452  040361  000004                  BIC     R3,4(R1)        ;CLEAR TCR BIT FOR LINE
007456  012603                          MOV     (SP)+,R3        ;RESTORE R3
007460  000741                          BR      1$
007462                          4$:
007462  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
007464  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
007466  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
007470  000002                          RTI

                                ;*******************************************************************************
                                ;PROG1  RECEIVER INTERRUPT SERVICE ROUTINE
                                ;*******************************************************************************

007472                          RCVISR:
007472  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
```

```
                                                        MOV     R2,-(6)         ;PUSH R2 ON STACK
                                                        MOV     R3,-(6)         ;PUSH R3 ON STACK
                                                        MOV     R4,-(6)         ;PUSH R4 ON STACK
                                                        MOV     (R0)+,R1        ;GET RBUF ADDRESS
                                                1$:     MOV     (R1),R2         ;READ THE DATA
                                                        BPL     7$              ;BRANCH IF NO CHAR PRESENT
                                                        BIT     #70000,R2       ;CHECK FOR ERRORS
                                                        BEQ     2$              ;BRANCH IF OK
                                                        HLT+2                   ;RECEIVER ERROR
                                                                                ;R1=RBUF ADDRESS
                                                                                ;R2=CONTENTS OF RBUF
                                                                                ;BIT12=PARITY ERROR
                                                                                ;BIT13=FRAMING ERROR
                                                                                ;BIT14=UART OVERRUN
                                                2$:     BIC     #70000, R2      ;CLEAR ERROR BITS FOR SPEED
                                                        MOV     R2,     R4      ;DUP THE RBUF
                                                        CLRB    R4              ;CLEAR THE DATA
                                                        SWAB    R4              ;LINE # TO LOW BYTE
                                                        ASLB    R4              ;LINE # * 2, ALSO CLR CHAR PRESENT
                                                        ADD     (R0),R4         ;ADD OFFSET
                                                        MOVB    @RCVTAB(R4),R3  ;GET EXPECTED DATA
                                                        BIC     MASK(4),R3      ;MASK CHARACTER LENGTH
                                                        CMPB    R3,R2
                                                        BEQ     3$              ;BRANCH IF OK
                                                        BIC     #177400,R3      ;MAKE SURE UPPER BYTE CLEAR
                                                        HLT+3                   ;DATA ERROR
                                                                                ;R1=RBUF ADDRESS
                                                                                ;R2=CONTENTS OF RBUF(DATA)
                                                                                ;R3=EXPECTED DATA
                                                3$:     DECB    RCVCNT(4)
                                                        BEQ     4$
                                                        INC     RCVTAB(4)       ;UPDATE TABLE POINTER
                                                        BR      1$              ;CONTINUE
                                                4$:     TSTB    CNTTAB(4)       ;CHECK FOR DONE 8 MESSAGES
                                                        BMI     1$              ;BRANCH IF DONE (DON'T RESTART TRANSMITTER)
                                                        BEQ     1$              ;BRANCH IF DONE (DON'T RESTART TRANSMITTER)
                                                ;NOTE:  IF THE ABOVE LOCATION IS PATCHED TO 1402, THE FAST LINES WILL
                                                ;         CONTINUE TO RUN UNTIL ALL LINES HAVE COMPLETED 8 MESSAGES.
                                                ;         THE WAY IT IS CODED NOW, ALL LINES COMPLETE JUST 8 MESSAGES.
                                                        DECB    CNTTAB(4)
                                                5$:     MOV     @SWR,R3 ;GET SWITCH SETTINGS
                                                        BIC     #177770,R3      ;MASK SW<2:0>
                                                        ASL     R3
                                                        ASL     R3
                                                        MOV     ADRNIT(3),XMTTAB(4)
                                                        MOV     ADRNIT(3),RCVTAB(4)
                                                        MOVB    CNTNIT(3),XMTCNT(4)
                                                        MOVB    CNTNIT(3),RCVCNT(4)
                                                        SUB     (R0),R4
                                                        ASR     R4
                                                        CLR     R3
                                                        SEC
                                                6$:     ROL     R3
                                                        DEC     R4
                                                        BPL     6$
                                                        BIS     R3,2(R1)        ;SET TCR BIT FOR LINE
```

```
1032  007676  000702                        BR      1$
1033  007700                          7$:
1034  007700  012604                        MOV     (6)+,R4         ;POP STACK INTO R4
1035  007702  012603                        MOV     (6)+,R3         ;POP STACK INTO R3
1036  007704  012602                        MOV     (6)+,R2         ;POP STACK INTO R2
1037  007706  012601                        MOV     (6)+,R1         ;POP STACK INTO R1
1038  007710  012600                        MOV     (6)+,R0         ;POP STACK INTO R0
1039  007712  000002                        RTI
1040
1041                                  ;*****************************************************************
1042                                  ;PROG1  DATA TABLES
1043                                  ;*****************************************************************
1044
1045  007714  007760                  ADRNIT: BINARY                  ;SW<2:0>=0 BINARY COUNT PATTERN
1046  007716  000377                  CNTNIT: 377                     ;SIZE=256.
1047  007720  010360                          PHRASE                  ;SW<2:0>=1 "THE QUICK BROWN FOX..."
1048  007722  000100                          64.                     ;SIZE=64.
1049  007724  007752                          SIXBIT                  ;SW<2:0>=2 040 THRU 137
1050  007726  000106                          70.                     ;SIZE=70.
1051  007730  015360                          END
1052  007732  000001                          1
1053  007734  015760                          END+400
1054  007736  000001                          1
1055  007740  016360                          END+1000
1056  007742  000001                          1
1057  007744  016760                          END+1400
1058  007746  000001                          1
1059  007750  017360                          END+2000
1060  007752  005015  177777  177777  SIXBIT: .ASCII  <15><12><377><377><377><377>    ;CR-LF, FILLERS
1061  007760      040                 BINARY: .BYTE   40
1062  007761      041                         .BYTE   41
1063  007762      042                         .BYTE   42
1064  007763      043                         .BYTE   43
1065  007764      044                         .BYTE   44
1066  007765      045                         .BYTE   45
1067  007766      046                         .BYTE   46
1068  007767      047                         .BYTE   47
1069  007770      050                         .BYTE   50
1070  007771      051                         .BYTE   51
1071  007772      052                         .BYTE   52
1072  007773      053                         .BYTE   53
1073  007774      054                         .BYTE   54
1074  007775      055                         .BYTE   55
1075  007776      056                         .BYTE   56
1076  007777      057                         .BYTE   57
1077  010000      060                         .BYTE   60
1078  010001      061                         .BYTE   61
1079  010002      062                         .BYTE   62
1080  010003      063                         .BYTE   63
1081  010004      064                         .BYTE   64
1082  010005      065                         .BYTE   65
1083  010006      066                         .BYTE   66
1084  010007      067                         .BYTE   67
1085  010010      070                         .BYTE   70
1086  010011      071                         .BYTE   71
1087  010012      072                         .BYTE   72
```

```
 1088  010013      073                    .BYTE   73
 1089  010014      074                    .BYTE   74
 1090  010015      075                    .BYTE   75
 1091  010016      076                    .BYTE   76
 1092  010017      077                    .BYTE   77
 1093  010020      100                    .BYTE   100
 1094  010021      101                    .BYTE   101
 1095  010022      102                    .BYTE   102
 1096  010023      103                    .BYTE   103
 1097  010024      104                    .BYTE   104
 1098  010025      105                    .BYTE   105
 1099  010026      106                    .BYTE   106
 1100  010027      107                    .BYTE   107
 1101  010030      110                    .BYTE   110
 1102  010031      111                    .BYTE   111
 1103  010032      112                    .BYTE   112
 1104  010033      113                    .BYTE   113
 1105  010034      114                    .BYTE   114
 1106  010035      115                    .BYTE   115
 1107  010036      116                    .BYTE   116
 1108  010037      117                    .BYTE   117
 1109  010040      120                    .BYTE   120
 1110  010041      121                    .BYTE   121
 1111  010042      122                    .BYTE   122
 1112  010043      123                    .BYTE   123
 1113  010044      124                    .BYTE   124
 1114  010045      125                    .BYTE   125
 1115  010046      126                    .BYTE   126
 1116  010047      127                    .BYTE   127
 1117  010050      130                    .BYTE   130
 1118  010051      131                    .BYTE   131
 1119  010052      132                    .BYTE   132
 1120  010053      133                    .BYTE   133
 1121  010054      134                    .BYTE   134
 1122  010055      135                    .BYTE   135
 1123  010056      136                    .BYTE   136
 1124  010057      137                    .BYTE   137
 1125  010060      140                    .BYTE   140
 1126  010061      141                    .BYTE   141
 1127  010062      142                    .BYTE   142
 1128  010063      143                    .BYTE   143
 1129  010064      144                    .BYTE   144
 1130  010065      145                    .BYTE   145
 1131  010066      146                    .BYTE   146
 1132  010067      147                    .BYTE   147
 1133  010070      150                    .BYTE   150
 1134  010071      151                    .BYTE   151
 1135  010072      152                    .BYTE   152
 1136  010073      153                    .BYTE   153
 1137  010074      154                    .BYTE   154
 1138  010075      155                    .BYTE   155
 1139  010076      156                    .BYTE   156
 1140  010077      157                    .BYTE   157
 1141  010100      160                    .BYTE   160
 1142  010101      161                    .BYTE   161
 1143  010102      162                    .BYTE   162
```

```
1144  010103      163                         .BYTE   163
1145  010104      164                         .BYTE   164
1146  010105      165                         .BYTE   165
1147  010106      166                         .BYTE   166
1148  010107      167                         .BYTE   167
1149  010110      170                         .BYTE   170
1150  010111      171                         .BYTE   171
1151  010112      172                         .BYTE   172
1152  010113      173                         .BYTE   173
1153  010114      174                         .BYTE   174
1154  010115      175                         .BYTE   175
1155  010116      176                         .BYTE   176
1156  010117      177                         .BYTE   177
1157  010120      200                         .BYTE   200
1158  010121      201                         .BYTE   201
1159  010122      202                         .BYTE   202
1160  010123      203                         .BYTE   203
1161  010124      204                         .BYTE   204
1162  010125      205                         .BYTE   205
1163  010126      206                         .BYTE   206
1164  010127      207                         .BYTE   207
1165  010130      210                         .BYTE   210
1166  010131      211                         .BYTE   211
1167  010132      212                         .BYTE   212
1168  010133      213                         .BYTE   213
1169  010134      214                         .BYTE   214
1170  010135      215                         .BYTE   215
1171  010136      216                         .BYTE   216
1172  010137      217                         .BYTE   217
1173  010140      220                         .BYTE   220
1174  010141      221                         .BYTE   221
1175  010142      222                         .BYTE   222
1176  010143      223                         .BYTE   223
1177  010144      224                         .BYTE   224
1178  010145      225                         .BYTE   225
1179  010146      226                         .BYTE   226
1180  010147      227                         .BYTE   227
1181  010150      230                         .BYTE   230
1182  010151      231                         .BYTE   231
1183  010152      232                         .BYTE   232
1184  010153      233                         .BYTE   233
1185  010154      234                         .BYTE   234
1186  010155      235                         .BYTE   235
1187  010156      236                         .BYTE   236
1188  010157      237                         .BYTE   237
1189  010160      240                         .BYTE   240
1190  010161      241                         .BYTE   241
1191  010162      242                         .BYTE   242
1192  010163      243                         .BYTE   243
1193  010164      244                         .BYTE   244
1194  010165      245                         .BYTE   245
1195  010166      246                         .BYTE   246
1196  010167      247                         .BYTE   247
1197  010170      250                         .BYTE   250
1198  010171      251                         .BYTE   251
1199  010172      252                         .BYTE   252
```

```
1200  010173      253                     .BYTE   253
1201  010174      254                     .BYTE   254
1202  010175      255                     .BYTE   255
1203  010176      256                     .BYTE   256
1204  010177      257                     .BYTE   257
1205  010200      260                     .BYTE   260
1206  010201      261                     .BYTE   261
1207  010202      262                     .BYTE   262
1208  010203      263                     .BYTE   263
1209  010204      264                     .BYTE   264
1210  010205      265                     .BYTE   265
1211  010206      266                     .BYTE   266
1212  010207      267                     .BYTE   267
1213  010210      270                     .BYTE   270
1214  010211      271                     .BYTE   271
1215  010212      272                     .BYTE   272
1216  010213      273                     .BYTE   273
1217  010214      274                     .BYTE   274
1218  010215      275                     .BYTE   275
1219  010216      276                     .BYTE   276
1220  010217      277                     .BYTE   277
1221  010220      300                     .BYTE   300
1222  010221      301                     .BYTE   301
1223  010222      302                     .BYTE   302
1224  010223      303                     .BYTE   303
1225  010224      304                     .BYTE   304
1226  010225      305                     .BYTE   305
1227  010226      306                     .BYTE   306
1228  010227      307                     .BYTE   307
1229  010230      310                     .BYTE   310
1230  010231      311                     .BYTE   311
1231  010232      312                     .BYTE   312
1232  010233      313                     .BYTE   313
1233  010234      314                     .BYTE   314
1234  010235      315                     .BYTE   315
1235  010236      316                     .BYTE   316
1236  010237      317                     .BYTE   317
1237  010240      320                     .BYTE   320
1238  010241      321                     .BYTE   321
1239  010242      322                     .BYTE   322
1240  010243      323                     .BYTE   323
1241  010244      324                     .BYTE   324
1242  010245      325                     .BYTE   325
1243  010246      326                     .BYTE   326
1244  010247      327                     .BYTE   327
1245  010250      330                     .BYTE   330
1246  010251      331                     .BYTE   331
1247  010252      332                     .BYTE   332
1248  010253      333                     .BYTE   333
1249  010254      334                     .BYTE   334
1250  010255      335                     .BYTE   335
1251  010256      336                     .BYTE   336
1252  010257      337                     .BYTE   337
1253  010260      340                     .BYTE   340
1254  010261      341                     .BYTE   341
1255  010262      342                     .BYTE   342
```

```
  1256  010263     343                          .BYTE   343
  1257  010264     344                          .BYTE   344
  1258  010265     345                          .BYTE   345
  1259  010266     346                          .BYTE   346
  1260  010267     347                          .BYTE   347
  1261  010270     350                          .BYTE   350
  1262  010271     351                          .BYTE   351
  1263  010272     352                          .BYTE   352
  1264  010273     353                          .BYTE   353
  1265  010274     354                          .BYTE   354
  1266  010275     355                          .BYTE   355
  1267  010276     356                          .BYTE   356
  1268  010277     357                          .BYTE   357
  1269  010300     360                          .BYTE   360
  1270  010301     361                          .BYTE   361
  1271  010302     362                          .BYTE   362
  1272  010303     363                          .BYTE   363
  1273  010304     364                          .BYTE   364
  1274  010305     365                          .BYTE   365
  1275  010306     366                          .BYTE   366
  1276  010307     367                          .BYTE   367
  1277  010310     370                          .BYTE   370
  1278  010311     371                          .BYTE   371
  1279  010312     372                          .BYTE   372
  1280  010313     373                          .BYTE   373
  1281  010314     374                          .BYTE   374
  1282  010315     375                          .BYTE   375
  1283  010316     376                          .BYTE   376
  1284  010317     377                          .BYTE   377
  1285  010320     000                          .BYTE   0
  1286  010321     001                          .BYTE   1
  1287  010322     002                          .BYTE   2
  1288  010323     003                          .BYTE   3
  1289  010324     004                          .BYTE   4
  1290  010325     005                          .BYTE   5
  1291  010326     006                          .BYTE   6
  1292  010327     007                          .BYTE   7
  1293  010330     010                          .BYTE   10
  1294  010331     011                          .BYTE   11
  1295  010332     012                          .BYTE   12
  1296  010333     013                          .BYTE   13
  1297  010334     014                          .BYTE   14
  1298  010335     015                          .BYTE   15
  1299  010336     016                          .BYTE   16
  1300  010337     017                          .BYTE   17
  1301  010340     020                          .BYTE   20
  1302  010341     021                          .BYTE   21
  1303  010342     022                          .BYTE   22
  1304  010343     023                          .BYTE   23
  1305  010344     024                          .BYTE   24
  1306  010345     025                          .BYTE   25
  1307  010346     026                          .BYTE   26
  1308  010347     027                          .BYTE   27
  1309  010350     030                          .BYTE   30
  1310  010351     031                          .BYTE   31
  1311  010352     032                          .BYTE   32
```

```
1312  010353      033                                   .BYTE   33
1313  010354      034                                   .BYTE   34
1314  010355      035                                   .BYTE   35
1315  010356      036                                   .BYTE   36
1316  010357      037                                   .BYTE   37
1317
1318  010360  005015  177777  177777  PHRASE: .ASCII  <15><12><377><377><377><377>
1319  010366  044124  020105  052521          .ASCIZ  "THE QUICK BROWN FOX JUMPED OVER 9,876,543,210.0 LAZY DOGS!"
1320  010374  041511  020113  051102
1321  010402  053517  020116  047506
1322  010410  020130  052512  050115
1323  010416  042105  047440  042526
1324  010424  020122  026071  033470
1325  010432  026066  032065  026063
1326  010440  030462  027060  020060
1327  010446  040514  054532  042040
1328  010454  043517  020523      000
1329          010462                .EVEN
```

```
1330
1331                                    ;*************************************************************************
1332                                    ;PROGRAM 2:     ON-LINE MULTI-ECHO EXERCISER
1333                                    ;               TRANSMITS THE LAST CHARACTER RECEIVED ON ITS RESPECTIVE
1334                                    ;               LINE.  A CARRIAGE RETURN AND LINE FEED ARE INSERTED
1335                                    ;               EVERY 72 CHARACTERS.
1336                                    ;*************************************************************************
1337
1338  010462  000005                    PROG2:  RESET                           ;CLEAR OUT THE WORLD
1339  010464  012706  001200                    MOV     #ICNT,  SP              ;RESET THE STACK POINTER
1340  010470  052737  000340  177776            BIS     #340,   @#PS            ;PROCESSOR TO LEVEL 7
1341  010476  012701  001306                    MOV     #XMTTAB,R1              ;FIRST TABLE POINTER
1342  010502  012702  001400                    MOV     #1400,  R2                      ;LENGTH OF TABLES (WORDS)
1343  010506  005021                    1$:     CLR     (R1)+                   ;CLEAR THE TABLE
1344  010510  005302                            DEC     R2
1345  010512  001375                            BNE     1$
1346  010514  012702  000400                    MOV     #400,R2                 ;LENGTH OF MASK/COUNT TABLE
1347  010520  005201                    2$:     INC     R1                      ;SKIP MASK
1348  010522  105021                            CLRB    (R1)+                   ;CLEAR COUNT
1349  010524  005302                            DEC     R2
1350  010526  001374                            BNE     2$
1351
1352                                    ;ROUTINE TO INITIALIZE ALL DJ11'S AND THEIR ISR'S:
1353                                    ;SET UP ALL INTERRUPT VECTORS
1354                                    ;SET UP DEVICE ADDRESSES IN LINKER ROUTINES
1355                                    ;SET CSR'S EVERYTHING ENABLED
1356                                    ;SET TCR'S, ALL LINES ENABLED
1357
1358  010530  005000                    P2INIT: CLR     R0
1359  010532  013701  001272                    MOV     DEVADR, R1
1360  010536  013702  001274                    MOV     VECADR, R2
1361  010542  012703  010720                    MOV     #R2SR0,R3               ;SET UP POINTER TO LINKERS
1362  010546  010322                    1$:     MOV     R3,     (R2)+           ;SET UP RECEIVER INTERUPT VECTOR
1363  010550  013722  001302                    MOV     RCVLVL, (R2)+
1364  010554  022323                            CMP     (R3)+,  (R3)+           ;ADD 4 TO R3
1365  010556  010113                            MOV     R1,     (R3)            ;ADDRESS OF CSR
1366  010560  062723  000002                    ADD     #2,     (R3)+           ;ADDRESS OF RBUF
1367  010564  005723                            TST     (R3)+
1368  010566  010322                            MOV     R3,     (R2)+           ;SET UP TRANSMITTER INTERUPT VECTOR
1369  010570  013722  001304                    MOV     XMTLVL, (R2)+
1370  010574  022323                            CMP     (R3)+,  (R3)+
1371  010576  010123                            MOV     R1,     (R3)+           ;ADDRESS OF CSR
1372  010600  012721  050501                    MOV     #50501, (R1)+           ;SET UP CSR
1373                                                                            ;BIT0 = RECEIVER ENABLE
1374                                                                            ;BIT6 = RECEIVER INTERUPT ENABLE
1375                                                                            ;BIT8 = TRANSMITTER SCAN ENABLE
1376                                                                            ;BIT12 = STATUS ENABLE
1377                                                                            ;BIT14 = TRANSMITTER INTERUPT ENABLE
1378  010604  005721                            TST     (R1)+
1379  010606  006300                            ASL     R0                      ;UNIT # * 2
1380  010610  016011  001222                    MOV     SVSW0(R0),(R1)          ;SET TCR BITS FOR SELECTED LINES
1381  010614  006200                            ASR     R0                      ;RESET UNIT COUNTER
1382  010616  012737  000001  001262            MOV     #1,     MARK            ;SET UP MARKER
1383  010624  012304                            MOV     (R3)+,  R4              ;SET UP OFFSET TO TABLES
1384  010626  033711  001262            4$:     BIT     MARK,   (R1)            ;CHECK FOR LINE SELECTED
1385  010632  001406                            BEQ     5$
```

```
1386  010634  012764  015271  001306          MOV     #MSGP2, XMTTAB(4) ;SET UP XMTR TABLE
1387  010642  012764  000045  003306          MOV     #45,    XMTCNT(4) ;SET UP COUNT
1388  010650  005724                  5$:     TST     (R4)+             ;INC OFFSET TO NEXT LINE
1389  010652  006337  001262                  ASL     MARK
1390  010656  103363                          BCC     4$
1391  010660  022121                          CMP     (R1)+,  (R1)+     ;ADD 4
1392  010662  005200                          INC     R0
1393  010664  020037  001270                  CMP     R0,     UNITS
1394  010670  001326                          BNE     1$
1395  010672  042737  000140  177776          BIC     #140,   @#PS      ;LOWER PROCESSOR PRIORITY
1396
1397                                  ;*************************************************************************
1398                                  ;PROG2  FOREGROUND PROGRAM TO READ/WRITE MEMORY
1399                                  ;*************************************************************************
1400
1401  010700  012700  020000          FORP2:  MOV     #20000,R0         ;TOP OF 4K BANK OF MEMORY
1402  010704  000241                          CLC
1403  010706  005540                  1$:     ADC     -(R0)             ;FAST READ/WRITE TO MEMORY
1404  010710  001376                          BNE     1$                ;RAPID REPEAT
1405  010712  005700                          TST     R0                ;CHECK FOR LOC 0
1406  010714  001374                          BNE     1$                ;BRANCH IF MORE MEMORY
1407  010716  000770                          BR      FORP2             ;LOOP FOR EVER!
1408
1409                                  ;*************************************************************************
1410                                  ;PROG2 LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
1411                                  ;*************************************************************************
1412
1413  010720  004037  011512          R2SR0:  JSR     R0,P2RISR
1414  010724  160012  000000                  .WORD   <160012+<0*10>>,<0*40>
1415  010730  004037  011320          X2SR0:  JSR     R0,P2XISR
1416  010734  160020  000000                  .WORD   <160020+<0*10>>,<0*40>
1417  010740  004037  011512          R2SR1:  JSR     R0,P2RISR
1418  010744  160022  000040                  .WORD   <160012+<1*10>>,<1*40>
1419  010750  004037  011320          X2SR1:  JSR     R0,P2XISR
1420  010754  160030  000040                  .WORD   <160020+<1*10>>,<1*40>
1421  010760  004[   011512          R2SR2:  JSR     R0,P2RISR
1422  010764  160032  000100                  .WORD   <160012+<2*10>>,<2*40>
1423  010770  004037  011320          X2SR2:  JSR     R0,P2XISR
1424  010774  160040  000[00                  .WORD   <160020+<2*10>>,<2*40>
1425  011000  004037  011[12          R2SR3:  JSR     R0,P2RISR
1426  011004  160042  000140                  .WORD   <160012+<3*10>>,<3*40>
1427  011010  004037  011320          X2SR3:  JSR     R0,P2XISR
1428  011014  160050  000140                  .WORD   <160020+<3*10>>,<3*40>
1429  011020  004037  011512          R2SR4:  JSR     R0,P2RISR
1430  011024  160052  000200                  .WORD   <160012+<4*10>>,<4*40>
1431  011030  004037  011320          X2SR4:  JSR     R0,P2XISR
1432  011034  160060  000200                  .WORD   <160020+<4*10>>,<4*40>
1433  011040  004037  011512          R2SR5:  JSR     R0,P2RISR
1434  011044  160062  000240                  .WORD   <160012+<5*10>>,<5*40>
1435  011050  004037  011320          X2SR5:  JSR     R0,P2XISR
1436  011054  160070  000240                  .WORD   <160020+<5*10>>,<5*40>
1437  011060  004037  011512          R2SR6:  JSR     R0,P2RISR
1438  011064  160072  000300                  .WORD   <160012+<6*10>>,<6*40>
1439  011070  004037  011320          X2SR6:  JSR     R0,P2XISR
1440  011074  160100  000300                  .WORD   <160020+<6*10>>,<6*40>
1441  011100  004037  011512          R2SR7:  JSR     R0,P2RISR
```

```
1442  011104  160102  000340                    .WORD   <160012+<7*10>>,<7*40>
1443  011110  004037  011320          X2SR7:    JSR     R0,P2XISR
1444  011114  160110  000340                    .WORD   <160020+<7*10>>,<7*40>
1445  011120  004037  011512          R2SR10:   JSR     R0,P2RISR
1446  011124  160112  000400                    .WORD   <160012+<10*10>>,<10*40>
1447  011130  004037  011320          X2SR10:   JSR     R0,P2XISR
1448  011134  160120  000400                    .WORD   <160020+<10*10>>,<10*40>
1449  011140  004037  011512          R2SR11:   JSR     R0,P2RISR
1450  011144  160122  000440                    .WORD   <160012+<11*10>>,<11*40>
1451  011150  004037  011320          X2SR11:   JSR     R0,P2XISR
1452  011154  160130  000440                    .WORD   <160020+<11*10>>,<11*40>
1453  011160  004037  011512          R2SR12:   JSR     R0,P2RISR
1454  011164  160132  000500                    .WORD   <160012+<12*10>>,<12*40>
1455  011170  004037  011320          X2SR12:   JSR     R0,P2XISR
1456  011174  160140  000500                    .WORD   <160020+<12*10>>,<12*40>
1457  011200  004037  011512          R2SR13:   JSR     R0,P2RISR
1458  011204  160142  000540                    .WORD   <160012+<13*10>>,<13*40>
1459  011210  004037  011320          X2SR13:   JSR     R0,P2XISR
1460  011214  160150  000540                    .WORD   <160020+<13*10>>,<13*40>
1461  011220  004037  011512          R2SR14:   JSR     R0,P2RISR
1462  011224  160152  000600                    .WORD   <160012+<14*10>>,<14*40>
1463  011230  004037  011320          X2SR14:   JSR     R0,P2XISR
1464  011234  160160  000600                    .WORD   <160020+<14*10>>,<14*40>
1465  011240  004037  011512          R2SR15:   JSR     R0,P2RISR
1466  011244  160162  000640                    .WORD   <160012+<15*10>>,<15*40>
1467  011250  004037  011320          X2SR15:   JSR     R0,P2XISR
1468  011254  160170  000640                    .WORD   <160020+<15*10>>,<15*40>
1469  011260  004037  011512          R2SR16:   JSR     R0,P2RISR
1470  011264  160172  000700                    .WORD   <160012+<16*10>>,<16*40>
1471  011270  004037  011320          X2SR16:   JSR     R0,P2XISR
1472  011274  160200  000700                    .WORD   <160020+<16*10>>,<16*40>
1473  011300  004037  011512          R2SR17:   JSR     R0,P2RISR
1474  011304  160202  000740                    .WORD   <160012+<17*10>>,<17*40>
1475  011310  004037  011320          X2SR17:   JSR     R0,P2XISR
1476  011314  160210  000740                    .WORD   <160020+<17*10>>,<17*40>
1477
1478                                ;*****************************************************************************
1479                                ;PROG2  TRANSMITTER INTERRUPT SERVICE ROUTINE
1480                                ;*****************************************************************************
1481
1482  011320                        P2XISR:
1483  011320  010146                       MOV     R1,-(6)         ;PUSH R1 ON STACK
1484  011322  010246                       MOV     R2,-(6)         ;PUSH R2 ON STACK
1485  011324  012001                       MOV     (R0)+,R1
1486  011326  005711               1$:     TST     (R1)            ;CHECK FOR TRANS READY
1487  011330  100064                       BPL     4$
1488  011332  116102  000007              MOVB    7(R1),R2        ;GET LINE NO.
1489  011336  006302                       ASL     R2
1490  011340  061002                       ADD     (R0),R2
1491  011342  105762  003306              TSTB    XMTCNT(2)       ;TST FOR ZERO
1492  011346  001413                       BEQ     2$              ;GET OUT
1493  011350  117261  001306  000006      MOVB    @XMTTAB(2),6(R1);SEND A CHARACTER
1494  011356  105362  003306              DECB    XMTCNT(2)       ;COUNT CHARACTERS
1495  011362  105762  004307              TSTB    CNTTAB(2)       ;CHECK FOR MESSAGE OR DATA
1496  011366  001357                       BNE     1$              ;BRANCH IF DATA
1497  011370  005262  001306              INC     XMTTAB(2)       ;UPDATE TABLE POINTER
```

```
1498  011374  000754                          BR      1$
1499  011376  105162  004307          2$:     COMB    CNTTAB(2)           ;CHANGE FLAG
1500  011402  001430                          BEQ     3$                  ;BRANCH IF WAS DATA
1501  011404  012762  002306  001306          MOV     #RCVTAB,XMTTAB(2)   ;SET UP POINTER TO RECEIVER TABLE
1502  011412  060262  001300                  ADD     R2,     XMTTAB(2)   ;F O OFFSET
1503  011416  112762  000110  003306          MOVB    #72.,   XMTCNT(2)   ;COUNT 72. CHARACTERS TO THE LINE
1504  011424  105762  002306                  TSTB    RCVTAB(2)           ;CHECK FOR A BREAK
1505  011430  001336                          BNE     1$                  ;BRANCH IF REAL DATA
1506  011432  161002                          SUB     (R0),   R2          ;RECOVER LINE NUMBER
1507  011434  006202                          ASR     R2
1508  011436  005037  001262                  CLR     MARK                ;SET UP MARKER
1509  011442  000261                          SEC
1510  011444  006137  001262          5$:     ROL     MARK                ;MOVE MARKER
1511  011450  005302                          DEC     R2                  ;COUNT LINES
1512  011452  100374                          BPL     5$                  ;BRANCH IF MORE
1513  011454  043761  001262  000004          BIC     MARK,   4(R1)       ;CLEAR TCR BIT
1514  011462  000721                          BR      1$                  ;CONTINUE
1515  011464  012762  015056  001306  3$:     MOV     #RETURN,XMTTAB(2)   ;TYPE CARRIAGE RETURN, LINE FEED
1516  011472  112762  000002  003306          MOVB    #2,     XMTCNT(2)   ;COUNTER OF 2 CHARACTERS
1517  011500  000712                          BR      1$
1518  011502                          4$:
1519  011502  012602                          MOV     (6)+,R2             ;POP STACK INTO R2
1520  011504  012601                          MOV     (6)+,R1             ;POP STACK INTO R1
1521  011506  012600                          MOV     (6)+,R0             ;POP STACK INTO R0
1522  011510  000002                          RTI
1523
1524                          ;*****************************************************************************
1525                          ;PROG2  RECEIVER INTERRUPT SERVICE ROUTINE
1526                          ;*****************************************************************************
1527
1528  011512                  P2RISR:
1529  011512  010146                  MOV     R1,-(6)             ;PUSH R1 ON STACK
1530  011514  010246                  MOV     R2,-(6)             ;PUSH R2 ON STACK
1531  011516  010346                  MOV     R3,-(6)             ;PUSH R3 ON STACK
1532  011520  012001                  MOV     (R0)+,R1            ;GET RBUF ADDRESS
1533  011522  011102          1$:     MOV     (R1),R2             ;READ THE DATA
1534  011524  100053                  BPL     7$                  ;BRANCH IF NO CHAR PRESENT
1535  011526  032702  070000        BIT     #70000,R2           ;CHECK FOR ERRORS
1536  011532  001402                  BEQ     2$                  ;BRANCH IF OK
1537  011534  104002                  HLT+2                       ;RECEIVER ERROR
1538                                                              ;R1=RBUF ADDRESS
1539                                                              ;R2=CONTENTS OF RBUF
1540                                                              ;BIT12=PARITY ERROR
1541                                                              ;BIT13=FRAMING ERROR
1542                                                              ;BIT14=UART OVERRUN
1543  011536  000771                  BR      1$                  ;FORGET THE DATA
1544
1545  011540  010203          2$:     MOV     R2,     R3          ;DUP THE RBUF
1546  011542  105003                  CLRB    R3                  ;CLEAR THE DATA
1547  011544  000303                  SWAB    R3                  ;LINE # TO LOW BYTE
1548  011546  106303                  ASLB    R3                  ;LINE # * 2, ALSO CLR CHAR PRESENT
1549  011550  061003                  ADD     (R0),R3             ;ADD OFFSET
1550  011552  136302  004306          BITB    MASK(3),R2          ;CHECK CHARACTER LENGTH
1551  011556  00.401                  BEQ     3$                  ;BRANCH IF OK
1552  011560  104002                  HLT+2                       ;CHARACTER LENGTH ERROR
1553                                                              ;R1=RBUF ADDRESS
```

```
1554
1555  011562  105763  002306          3$:   TSTB   RCVTAB(3)              ;R2=CONTENTS OF RBUF(DATA)
1556  011566  001017                        BNE    5$                     ;CHECK FOR BREAK
1557  011570  110263  002306                MOVB   R2,     RCVTAB(3)       ;BRANCH IF REAL DATA
1558  011574  161003                        SUB    (R0),   R3             ;SAVE THE DATA
1559  011576  006203                        ASR    R3                     ;RECOVER LINE NUMBER
1560  011600  005037  001262                CLR    MARK                   ;SET UP MARKER
1561  011604  000261                        SEC
1562  011606  006137  001262          4$:   ROL    MARK                   ;UPDATE MARKER
1563  011612  005303                        DEC    R3                     ;COUNT LINES
1564  011614  100374                        BPL    4$                     ;BRANCH IF MORE
1565  011616  053761  001262  000002        BIS    MARK,   2(R1)          ;SET TCR BIT
1566  011624  000736                        BR     1$                     ;CONTINUE
1567
1568  011626  110263  002306          5$:   MOVB   R2,     RCVTAB(3)       ;SAVE THE DATA
1569  011632  105163  004307                COMB   CNTTAB(3)              ;SET MESSAGE FLAG
1570  011636  012763  015056  001306        MOV    #RETURN,XMTTAB(3)       ;TYPE CARRIAGE RETURN, LINE FEED
1571  011644  112763  000002  003306        MOVB   #2,     XMTCNT(3)       ;MESSAGE LENGTH
1572  011652  000723                        BR     1$
1573  011654                          7$:
1574  011654  012603                        MOV    (6)+,R3                ;POP STACK INTO R3
1575  011656  012602                        MOV    (6)+,R2                ;POP STACK INTO R2
1576  011660  012601                        MOV    (6)+,R1                ;POP STACK INTO R1
1577  011662  012600                        MOV    (6)+,R0                ;POP STACK INTO R0
1578  011664  000002                        RTI
```

```
                                ;*****************************************************************
                                ;PROGRAM 3:     ECHO EXERCISER
                                ;*****************************************************************

011666  000005                  PROG3:  RESET                   ;CLEAR OUT THE WORLD
011670  012706  001200                  MOV     #ICNT,  SP      ;RESET THE STACK POINTER
011674  052737  000340  177776          BIS     #340,   @#PS    ;PROCESSOR TO LEVEL 7
011702  012701  001306                  MOV     #XMTTAB,R1      ;FIRST TABLE POINTER
011706  012702  001400                  MOV     #1400,  R2      ;LENGTH OF TABLES (WORDS)
011712  005021                  1$:     CLR     (R1)+           ;CLEAR THE TABLE
011714  005302                          DEC     R2
011716  001375                          BNE     1$
011720  012702  000400                  MOV     #400,R2         ;LENGTH OF MASK/COUNT TABLE
011724  005201                  2$:     INC     R1              ;SKIP MASK
011726  105021                          CLRB    (R1)+           ;CLEAR COUNT
011730  005302                          DEC     R2
011732  001374                          BNE     2$
011734  013700  001264                  MOV     BUFSIZ, R0      ;DUP BUFSIZ
011740  006200                          ASR     R0
011742  040037  001264          BUFNIT: BIC     R0,     BUFSIZ  ;CLEAR ALL BUT MSB OF BUFSIZ
011746  006200                          ASR     R0              ;MASK ALL LOW ORDER BITS
011750  001374                          BNE     BUFNIT
011752  013737  001264  001266          MOV     BUFSIZ, BUFMSK  ;SET UP BUFMSK
011760  005337  001266                  DEC     BUFMSK
011764  012705  015360                  MOV     #END,   R5      ;SET UP BUFFER POINTER
011770  063705  001266                  ADD     BUFMSK, R5      ;PUT IT ON A BOUNDARY
011774  043705  001266                  BIC     BUFMSK, R5

                                ;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

012000  005000                  P3INIT: CLR     R0
012002  013701  001272                  MOV     DEVADR, R1
012006  013702  001274                  MOV     VECADR, R2
012012  012703  012274                  MOV     #R3SR0,R3       ;SET UP POINTER TO LINKERS
012016  010322                  1$:     MOV     R3,     (R2)+   ;SET UP RECEIVER INTERUPT VECTOR
012020  013722  001302                  MOV     RCVLVL, (R2)+
012024  022323                          CMP     (R3)+,  (R3)+   ;ADD 4 TO R3
012026  010113                          MOV     R1,     (R3)    ;ADDRESS OF CSR
012030  062723  000002                  ADD     #2,     (R3)+   ;ADDRESS OF RBUF
012034  005723                          TST     (R3)+
012036  010322                          MOV     R3,     (R2)+   ;SET UP TRANSMITTER INTERUPT VECTOR
012040  013722  001304                  MOV     XMTLVL, (R2)+
012044  022323                          CMP     (R3)+,  (R3)+
012046  010123                          MOV     R1,     (R3)+   ;ADDRESS OF CSR
012050  012721  050400                  MOV     #50400, (R1)+   ;SET UP CSR, TRANSMITTER ONLY
                                                                ;BIT8 = TRANSMITTER SCAN ENABLE
                                                                ;BIT12 = STATUS ENABLE
                                                                ;BIT14 = TRANSMITTER INTERUPT ENABLE
012054  005721                          TST     (R1)+
012056  006300                          ASL     R0              ;UNIT # * 2
012060  016011  001222                  MOV     SVSW0(R0),(R1)  ;SET TCR BITS FOR SELECTED LINES
```

```
1635  012064  006200                        ASR     R0              ;RESET UNIT COUNTER
1636  012066  012737  000001  001262        MOV     #1,     MARK    ;SET UP MARKER
1637  012074  012304                        MOV     (R3)+,  R4      ;SET UP OFFSET TO TABLES
1638  012076  010246                        MOV     R2,-(6)         ;PUSH R2 ON STACK
1639  012100  010346                        MOV     R3,-(6)         ;PUSH R3 ON STACK
1640  012102  033711  001262        2$:     BIT     MARK,   (R1)    ;CHECK FOR LINE SELECTED
1641  012106  001420                        BEQ     6$
1642  012110  010564  001306                MOV     R5, XMTTAB(4)   ;SET UP HEADER MESSAGE
1643  012114  010564  002306                MOV     R5, RCVTAB(4)   ;SET UP RECEIVER TABLE
1644  012120  013702  001264                MOV     BUFSIZ, R2      ;SET UP COUNTER
1645  012124  012703  015337                MOV     #MSGP3, R3      ;SET UP MESSAGE POINTER
1646  012130  112325                3$:     MOVB    (R3)+,  (R5)+   ;MOVE MESSAGE INTO BUFFER
1647  012132  001404                        BEQ     5$              ;BRANCH IF END OF MESSAGE
1648  012134  005302                        DEC     R2              ;COUNT BUFFER SIZE
1649  012136  001374                        BNE     3$              ;BRANCH IF MORE
1650  012140  000403                        BR      6$              ;BRANCH IF DONE
1651  012142  105025                4$:     CLRB    (R5)+           ;CLEAR REST OF BUFFER
1652  012144  005302                5$:     DEC     R2              ;COUNT BUFFER SIZE
1653  012146  001375                        BNE     4$              ;BRANCH IF MORE
1654  012150  005724                6$:     TST     (R4)+           ;INC OFFSET TO NEXT LINE
1655  012152  006337  001262                ASL     MARK
1656  012156  103351                        BCC     2$
1657  012160  012603                        MOV     (6)+,R3         ;POP STACK INTO R3
1658  012162  012602                        MOV     (6)+,R2         ;POP STACK INTO R2
1659  012164  022121                        CMP     (R1)+,  (R1)+   ;ADD 4
1660  012166  005200                        INC     R0
1661  012170  020037  001270                CMP     R0,     UNITS
1662  012174  001310                        BNE     1$
1663  012176  042737  000140  177776        BIC     #140,   @#PS    ;LOWER PROCESSOR PRIORITY
1664
1665                                ;*****************************************************************************
1666                                ;PROG3  FOREGROUND PROGRAM TO START RECEIVERS, THEN EXERCISE MEMORY.
1667                                ;*****************************************************************************
1668
1669  012204  012701  001306        FORP3:  MOV     #XMTTAB,R1
1670  012210  012702  000400                MOV     #400,R2
1671  012214  005711                1$:     TST     (R1)            ;CHECK FOR XMTR TABLE CLR
1672  012216  001376                        BNE     1$              ;BRANCH IF NOT
1673  012220  062701  000002                ADD     #2,R1           ;GO TO NEXT LINE ENTRY
1674  012224  005302                        DEC     R2              ;COUNT LINES
1675  012226  001372                        BNE     1$              ;BRANCH IF MORE LINES
1676  012230  013700  001270                MOV     UNITS,  R0      ;SET UP UNIT COUNTER
1677  012234  013701  001272                MOV     DEVADR, R1      ;AND DEVICE ADDRESS POINTER
1678  012240  052711  000101        2$:     BIS     #101,   (R1)    ;SET RECEIVER ENABLES OF CSR
1679                                                                ;BIT0 = RECEIVER ENABLE
1680                                                                ;BIT6 = RCV INTERRUPT ENABLE
1681  012244  062701  000010                ADD     #10,    R1      ;UPDATE TO NEXT DJ11
1682  012250  005300                        DEC     R0              ;COUNT DJ11'S
1683  012252  001372                        BNE     2$
1684  012254  012700  020000        MEMX3:  MOV     #20000,R0       ;TOP OF 4K BANK OF MEMORY
1685  012260  000241                        CLC
1686  012262  005540                1$:     ADC     -(R0)           ;FAST READ/WRITE TO MEMORY
1687  012264  001376                        BNE     1$              ;RAPID REPEAT
1688  012266  005700                        TST     R0              ;CHECK FOR LOC 0
1689  012270  001374                        BNE     1$              ;BRANCH IF MORE MEMORY
1690  012272  000770                        BR      MEMX3           ;LOOP FOR EVER!
```

```
1691
1692                                    ;*****************************************************************************
1693                                    ;PROG3  LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
1694                                    ;*****************************************************************************
1695
1696  012274  004037  013024   R3SR0:  JSR     R0,P3RISR
1697  012300  160012  000000           .WORD   <160012+<0*10>>,<0*40>
1698  012304  004037  012674   X3SR0:  JSR     R0,P3XISR
1699  012310  160020  000000           .WORD   <160020+<0*10>>,<0*40>
1700  012314  004037  013024   R3SR1:  JSR     R0,P3RISR
1701  012320  160022  000040           .WORD   <160012+<1*10>>,<1*40>
1702  012324  004037  012674   X3SR1:  JSR     R0,P3XISR
1703  012330  160030  000040           .WORD   <160020+<1*10>>,<1*40>
1704  012334  004037  013024   R3SR2:  JSR     R0,P3RISR
1705  012340  160032  000100           .WORD   <160012+<2*10>>,<2*40>
1706  012344  004037  012674   X3SR2:  JSR     R0,P3XISR
1707  012350  160040  000100           .WORD   <160020+<2*10>>,<2*40>
1708  012354  004037  013024   R3SR3:  JSR     R0,P3RISR
1709  012360  160042  000140           .WORD   <160012+<3*10>>;<3*40>
1710  012364  004037  012674   X3SR3:  JSR     R0,P3XISR
1711  012370  160050  000140           .WORD   <160020+<3*10>>,<3*40>
1712  012374  004037  013024   R3SR4:  JSR     R0,P3RISR
1713  012400  160052  000200           .WORD   <160012+<4*10>>,<4*40>
1714  012404  004037  012674   X3SR4:  JSR     R0,P3XISR
1715  012410  160060  000200           .WORD   <160020+<4*10>>,<4*40>
1716  012414  004037  013024   R3SR5:  JSR     R0,P3RISR
1717  012420  160062  000240           .WORD   <160012+<5*10>>,<5*40>
1718  012424  004037  012674   X3SR5:  JSR     R0,P3XISR
1719  012430  160070  000240           .WORD   <160020+<5*10>>,<5*40>
1720  012434  004037  013024   R3SR6:  JSR     R0,P3RISR
1721  012440  160072  000300           .WORD   <160012+<6*10>>,<6*40>
1722  012444  004037  012674   X3SR6:  JSR     R0,P3XISR
1723  012450  160100  000300           .WORD   <160020+<6*10>>,<6*40>
1724  012454  004037  013024   R3SR7:  JSR     R0,P3RISR
1725  012460  160102  000340           .WORD   <160012+<7*10>>,<7*40>
1726  012464  004037  012674   X3SR7:  JSR     R0,P3XISR
1727  012470  160110  000340           .WORD   <160020+<7*10>>,<7*40>
1728  012474  004037  013024   R3SR10: JSR     R0,P3RISR
1729  012500  160112  000400           .WORD   <160012+<10*10>>,<10*40>
1730  012504  004037  012674   X3SR10: JSR     R0,P3XISR
1731  012510  160120  000400           .WORD   <160020+<10*10>>,<10*40>
1732  012514  004037  013024   R3SR11: JSR     R0,P3RISR
1733  012520  160122  000440           .WORD   <160012+<11*10>>,<11*40>
1734  012524  004037  012674   X3SR11: JSR     R0,P3XISR
1735  012530  160130  000440           .WORD   <160020+<11*10>>,<11*40>
1736  012534  004037  013024   R3SR12: JSR     R0,P3RISR
1737  012540  160132  000500           .WORD   <160012+<12*10>>,<12*40>
1738  012544  004037  012674   X3SR12: JSR     R0,P3XISR
1739  012550  160140  000500           .WORD   <160020+<12*10>>,<12*40>
1740  012554  004037  013024   R3SR13: JSR     R0,P3RISR
1741  012560  160142  000540           .WORD   <160012+<13*10>>,<13*40>
1742  012564  004037  012674   X3SR13: JSR     R0,P3XISR
1743  012570  160150  000540           .WORD   <160020+<13*10>>,<13*40>
1744  012574  004037  013024   R3SR14: JSR     R0,P3RISR
1745  012600  160152  000600           .WORD   <160012+<14*10>>,<14*40>
1746  012604  004037  012674   X3SR14: JSR     R0,P3XISR
```

```
1747  012610  160160  000600                .WORD   <160020+<14*10>>,<14*40>
1748  012614  004037  013024  R3SR15: JSR     R0,P3RISR
1749  012620  160162  000640                .WORD   <160012+<15*10>>,<15*40>
1750  012624  004037  012674  X3SR15: JSR     R0,P3XISR
1751  012630  160170  000640                .WORD   <160020+<15*10>>,<15*40>
1752  012634  004037  013024  R3SR16: JSR     R0,P3RISR
1753  012640  160172  000700                .WORD   <160012+<16*10>>,<16*40>
1754  012644  004037  012674  X3SR16: JSR     R0,P3XISR
1755  012650  160200  000700                .WORD   <160020+<16*10>>,<16*40>
1756  012654  004037  013024  R3SR17: JSR     R0,P3RISR
1757  012660  160202  000740                .WORD   <160012+<17*10>>,<17*40>
1758  012664  004037  012674  X3SR17: JSR     R0,P3XISR
1759  012670  160210  000740                .WORD   <160020+<17*10>>,<17*40>
1760
1761                        ;*****************************************************************
1762                        ;PROG3  TRANSMITTER INTERRUPT SERVICE ROUTINE
1763                        ;*****************************************************************
1764
1765  012674                P3XISR:
1766  012674  010146                MOV     R1,-(6)         ;PUSH R1 ON STACK
1767  012676  010246                MOV     R2,-(6)         ;PUSH R2 ON STACK
1768  012700  012001                MOV     (R0)+,R1
1769  012702  005711        1$:     TST     (R1)            ;CHECK FOR TRANS READY
1770  012704  100043                BPL     4$
1771  012706  116102  000007       MOVB    7(R1),R2        ;GET LINE NO.
1772  012712  006302                ASL     R2
1773  012714  061002                ADD     (R0),R2
1774  012716  117261  001306  000006  MOVB  @XMTTAB(2),6(R1);SEND A CHARACTER
1775  012724  105072  001306        CLRB    @XMTTAB(2)      ;CLR TABLE AFTER USE
1776  012730  005262  001306        INC     XMTTAB(2)       ;UPDATE TABLE POINTER
1777  012734  033762  001266  001306  BIT   BUFMSK, XMTTAB(2) ;CHECK FOR END OF BUFFER
1778  012742  001003                BNE     5$              ;BRANCH IF NOT
1779  012744  163762  001264  001306  SUB   BUFSIZ, XMTTAB(2) ;RAP ARROUND BUFFER
1780  012752  105772  001306  5$:   TSTB    @XMTTAB(2)      ;CHECK NEXT CHARACTER
1781  012756  001351                BNE     1$              ;BRANCH IF MORE DATA
1782  012760  010346        2$:     MOV     R3,-(SP)
1783  012762  005062  001306        CLR     XMTTAB(2)       ;CLEAR TABLE POINTER
1784  012766  161002                SUB     (R0),R2
1785  012770  006202                ASR     R2
1786  012772  005003                CLR     R3
1787  012774  000261                SEC
1788  012776  006103        3$:     ROL     R3
1789  013000  005302                DEC     R2
1790  013002  100375                BPL     3$
1791  013004  040361  000004       BIC     R3,4(R1)        ;CLEAR TCR BIT FOR LINE
1792  013010  012603                MOV     (SP)+,R3        ;RESTORE R3
1793  013012  000733                BR      1$
1794  013014        4$:
1795  013014  012602                MOV     (6)+,R2         ;POP STACK INTO R2
1796  013016  012601                MOV     (6)+,R1         ;POP STACK INTO R1
1797  013020  012600                MOV     (6)+,R0         ;POP STACK INTO R0
1798  013022  000002                RTI
1799
1800                        ;*****************************************************************
1801                        ;PROG3  RECEIVER INTERRUPT SERVICE ROUTINE
1802                        ;*****************************************************************
```

```
013024                                    P3RISR:
013024  010146                                    MOV     R1,-(6)           ;PUSH R1 ON STACK
013026  010246                                    MOV     R2,-(6)           ;PUSH R2 ON STACK
013030  010346                                    MOV     R3,-(6)           ;PUSH R3 ON STACK
013032  010446                                    MOV     R4,-(6)           ;PUSH R4 ON STACK
013034  012001                                    MOV     (R0)+,R1          ;GET RBUF ADDRESS
013036  011102                            1$:     MOV     (R1),R2           ;READ THE DATA
013040  100072                                    BPL     8$                ;BRANCH IF NO CHAR PRESENT
013042  032702  070000                            BIT     #70000,R2         ;CHECK FOR ERRORS
013046  001402                                    BEQ     2$                ;BRANCH IF OK
013050  104002                                    HLT+2                     ;RECEIVER ERROR
                                                                            ;R1=RBUF ADDRESS
                                                                            ;R2=CONTENTS OF RBUF
                                                                            ;BIT12=PARITY ERROR
                                                                            ;BIT13=FRAMING ERROR
                                                                            ;BIT14=UART OVERRUN
013052  000771                                    BR      1$                ;SKIP BAD DATA
013054  010204                            2$:     MOV     R2,     R4        ;DUP THE RBUF
013056  105004                                    CLRB    R4                ;CLEAR THE DATA
013060  000304                                    SWAB    R4                ;LINE # TO LOW BYTE
013062  106304                                    ASLB    R4                ;LINE # * 2, ALSO CLR CHAR PRESENT
013064  061004                                    ADD     (R0),R4           ;ADD OFFSET
013066  136402  004306                            BITB    MASK(4),R2        ;CHECK CHARACTER LENGTH
013072  001401                                    BEQ     3$                ;BRANCH IF OK
013074  104002                                    HLT+2                     ;CHARACTER LENGTH ERROR
                                                                            ;R1=RBUF ADDRESS
                                                                            ;R2=CONTENTS OF RBUF(DATA)
013076  005764  002306                    3$:     TST     RCVTAB(4)         ;CHECK FOR UNSELECTED LINE
013102  001002                                    BNE     4$                ;BRANCH IF OK
013104  104002                                    HLT+2                     ;RECEIVED DATA ON UNSELECTED LINE
                                                                            ;R1 = RBUF ADDRESS
                                                                            ;R2 = CONTENTS OF RBUF
013106  000753                                    BR      1$                ;IGNORE THE DATA
013110  105774  002306                    4$:     TSTB    @RCVTAB(4)        ;CHECK FOR DATA BUFFER FULL
013114  001403                                    BEQ     5$                ;BRANCH IF OK
013116  104002                                    HLT+2                     ;SOFTWARE DATA BUFFER OVERFLOW
                                                                            ;POSSIBLE TRANSMITTER PROBLEM
                                                                            ;R1 = RBUF ADDRESS
                                                                            ;R2 = CONTENTS OF RBUF
                                          ;NOTE:  IF THE ABOVE ERROR WAS DUE TO OVERLOAD, INCREASING THE CONTENTS
                                          ;       OF "BUFSIZ" MAY RECTIFY THE PROBLEM.
                                          ;       "BUFSIZ" MUST BE A MULTIPLE OF 2.
                                          ;       INCREASING IT MAY CAUSE THE BUFFERS TO OVERFLOW 4K.
013120  000137  011666                            JMP     PROG3             ;RESTART ON THIS TYPE ERROR

013124  005764  001306                    5$:     TST     XMTTAB(4)         ;CHECK FOR TRANSMITTER ACTIVE
013130  001414                                    BEQ     6$                ;BRANCH IF INACTIVE
013132  110274  002306                            MOVB    R2, @RCVTAB(4)    ;PUT THE DATA IN THE BUFFER
013136  005264  002306                            INC     RCVTAB(4)         ;UPDATE POINTER TO NEXT SPACE
013142  033764  001266  002306                    BIT     BUFMSK, RCVTAB(4) ;CHECK FOR END OF BUFFER
013150  001332                                    BNE     1$                ;BRANCH IF NOT
013152  163764  001264  002306                    SUB     BUFSIZ, RCVTAB(4) ;RAP ARROUND BUFFER
013160  000726                                    BR      1$
013162  043764  001266  002306            6$:     BIC     BUFMSK, RCVTAB(4) ;RESET TABLE POINTER
013170  016464  002306  001306                    MOV     RCVTAB(4),XMTTAB(4)
```

```
 1859  013176  110274  002306                MOVB    R2, @RCVTAB(4)
 1860  013202  161004                        SUB     (R0),R4
 1861  013204  006204                        ASR     R4
 1862  013206  005003                        CLR     R3
 1863  013210  000261                        SEC
 1864  013212  006103                7$:     ROL     R3
 1865  013214  005304                        DEC     R4
 1866  013216  100375                        BPL     7$
 1867  013220  050361  000002                BIS     R3,2(R1)        ;SET TCR BIT FOR LINE
 1868  013224  000704                        BR      1$
 1869  013226                        9$:
 1870  013226  012604                        MOV     (6)+,R4         ;POP STACK INTO R4
 1871  013230  012603                        MOV     (6)+,R3         ;POP STACK INTO R3
 1872  013232  012602                        MOV     (6)+,R2         ;POP STACK INTO R2
 1873  013234  012601                        MOV     (6)+,R1         ;POP STACK INTO R1
 1874  013236  012600                        MOV     (6)+,R0         ;POP STACK INTO R0
 1875  013240  000002                        RTI
```

```
1876  013242                                DONE:
1877  013242  004737  015006                      JSR     PC,     KBDINT
1878  013246  062737  000001  001206              ADD     #1,PCNT+2       ;ADD 1 TO THE PASS COUNT
1879  013254  005537  001204                      ADC     PCNT            ;MAKE IT DOUBLE PREC.
1880  013260  032777  002000  165722              BIT     #SW10,@SWR      ;RING THE BELL?
1881  013266  001004                              BNE     4$              ;NO!
1882  013270  000004  000007                      TYPE    ,BELL           ;RING THE BELL
1883  013274  000004  000177                      TYPE    ,177            ;TYPE A FILLER FOR 11/05
1884  013300  013700  000042          4$:         MOV     @#42,R0         ;GET MONITOR ADDRESS
1885  013304  001405                              BEQ     3$              ;IF NONE
1886  013306  000005                              RESET                   ;RESET AND
1887          013310                              $ENDAD = .
1888  013310  004710                              JSR     7,(0)           ;GO TO MONITOR
1889  013312  000240                              NOP                     ;SAVE ROOM
1890  013314  000240                              NOP                     ;FOR
1891  013316  000240                              NOP                     ;ACT11
1892  013320  000137  006322          3$:         JMP     RESTAR          ;RETURN
1893
1894  013324  000000                  .TBIT:      0                       ;T BIT FLAG
1895
1896                                  ;       $HLT            ERROR TYPEOUT HANDLER
1897
1898                                  ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
1899                                  ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
1900                                  ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
1901                                  ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUEMENT
1902                                  ;(HLT+3) WILL BE PLACED IN "HLTCT$:" FOR ADITIONAL TYPEOUTS.
1903
1904  013326  004737  015006          EMT$:       JSR     PC,     KBDINT
1905  013332  032777  002000  165650              BIT     #SW10,@SWR      ;BELL ON ERROR?
1906  013340  001402                              BEQ     1$              ;NO - SKIP
1907  013342  000004  000007                      TYPE    ,BELL           ;RING BELL
1908  013346  005237  001202          1$:         INC     ERRORS          ;COUNT THE NUMBER OF ERRORS
1909  013352  032777  020000  165630              BIT     #SW13,@SWR      ;SKIP TYPEOUT IF SET
1910  013360  001026                              BNE     2$              ;SKIP TYPEOUTS
1911  013362  000004  013366                      TYPE    .+2             ;.ASCIZ <15><12>
1912  013372  011637  013456                      MOV     (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
1913  013376  162737  000002  013456              SUB     #2,HLTADR       ;FUDGE ADDRESS
1914  013404  117737  000046  013454              MOVB    @HLTADR,HLTCT$  ;GET HLT ARGUEMENT
1915  013412  013705  013456                      MOV     HLTADR,TTY      ;TYPE HLTADR IN OCTAL
1916  013416  004737  014064                      JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1917  013422  000004  013426                      TYPE    .+2             ;.ASCIZ "  "
1918  013432  004737  013460                      JSR     PC,ERROR$       ;GO TO USER ERROR ROUTINE
1919  013436  005777  165546          2$:         TST     @SWR            ;HALT ON ERROR
1920  013442  100001                              BPL     .+4             ;SKIP IF CONTINUE
1921  013444  000000                              HALT                    ;HALT ON ERROR!
1922  013446  004737  015006                      JSR     PC,KBDINT
1923  013452  000002                              RTI                     ;RETURN
1924
1925  013454  000000                  HLTCT$:     0                       ;HLT ARGUEMENT
1926  013456  000000                  HLTADR:     0                       ;LAST HLT INSTRUCTION EXECUTED
1927
1928  013460  042737  007700  013502  ERROR$:     BIC     #7700,2$
1929  013466  105337  013454          1$:         DECB    HLTCT$
1930  013472  100411                              BMI     3$
1931  013474  062737  000100  013502              ADD     #100,2$
```

```
 1932  013502  010005                  2$:     MOV     %0,TTY          ;TYPE REGISTER X IN OCTAL
 1933  013504  004737  014064                  JSR     %7,PRINTR
 1934  013510  000004  015064                  TYPE,   SPACE
 1935  013514  000764                          BR      1$
 1936  013516  000207                  3$:     RTS     PC
```

```
                                ;SUBROUTINE TO SAVE INPUT AS OCTAL NUMBER

013520  012737  000001  014012  READIN: MOV     #1,INHRE
013526  004737  013666                  JSR     PC,     READ$   ;GO READ TTY UNTIL CR
013532  005037  014012                  CLR     INHRE
013536  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
013540  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
013542  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
013544  012501                          MOV     (R5)+,  R1
013546  012737  000020  014704          MOV     #20,CNT
013554  012702  014014                  MOV     #INPUT,R2
013560  122712  000120                  CMPB    #120,(R2)       ;CHECK FOR "P"
013564  001425                          BEQ     3$
013566  005011                          CLR     (R1)
013570  112203                  1$:     MOVB    (R2)+,R3
013572  120327  000015                  CMPB    R3,#15
013576  001420                          BEQ     3$              ;BRANCH WHEN DONE
013600  162703  000060                  SUB     #60,R3
013604  032703  177770                  BIT     #177770,R3
013610  001013                          BNE     3$              ;BRANCH IF BAD DATA
013612  006311                          ASL     (R1)
013614  103410                          BCS     2$
013616  006311                          ASL     (R1)
013620  103406                          BCS     2$
013622  006311                          ASL     (R1)
013624  103404                          BCS     2$
013626  050311                          BIS     R3,(R1)
013630  005337  014704                  DEC     CNT
013634  000755                          BR      1$
013636  000244                  2$:     CLZ                     ;MAKE SURE Z-BIT IS CLR
013640  013737  177776  013664  3$:     MOV     @#PS,   PSTEMP  ;SAVE CONDITION CODES
013646  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
013650  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
013652  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
013654  013737  013664  177776          MOV     PSTEMP, @#PS    ;RESTORE CONDITION CODES
013662  000205                          RTS     R5

013664  000000                  PSTEMP  0               ;TEMPORARY STORAGE FOR PS

013666  010346                  READ$:  MOV     R3,-(6)         ;SAVE R3
013670  012703  014014          1$:     MOV     #INPUT,R3       ;GET ADDRESS
013674  022703  014034          2$:     CMP     #INPUT+20,R3    ;BUFFER FULL?
013700  001415                          BEQ     4$              ;YES - TYPE "?"
013702  105737  177560                  TSTB    @#177560        ;WAIT FOR
013706  100375                          BPL     .-4             ;A CHARACTER
013710  113713  177562                  MOVB    @#177562,(3)    ;GET CHARACTER
013714  142713  000200                  BICB    #200,(3)        ;GET RID OF JUNK
013720  122713  000177                  CMPB    #177,(3)        ;IS IT A RUBOUT
013724  001403                          BEQ     4$              ;SKIP IF NOT
013726  122713  000025                  CMPB    #25,(3)
013732  001006                          BNE     3$
013734                          4$:
013734  000004  013740                  TYPE    .+2             ;.ASCIZ "?"<15><12>"= "
013746  000750                          BR      1$              ;ZAP THE BUFFER AND LOOP
013750  111337  014622          3$:     MOVB    (3),.TYPE       ;SET UP FOR TYPING
```

```
 1993  013754  000004  014622                  TYPE    ,.TYPE          ;ECHO IT
 1994  013760  122723  000015                  CMPB    #15,(3)+        ;CHECK FOR RETURN
 1995  013764  001343                          BNE     2$              ;LOOP IF NOT RETURN
 1996  013766  005737  014012                  TST     INHRE
 1997  013772  001401                          BEQ     5$
 1998  013774  000402                          BR      6$
 1999  013776  105063  177777          5$:     CLRB    -1(3)           ;ZAP RETURN (THE 15)
 2000  014002  000004  000012          6$:     TYPE    ,12             ;TYPE A LINE FEED
 2001  014006  012603                          MOV     (6)+,R3         ;RESTORE R3
 2002  014010  000207                          RTS     PC              ;RETURN
 2003
 2004  014012  000000                  INHRE:  0
 2005  014014  000020                  INPUT:  .BLKW   20              ;TTY INPUT AREA
```

```
2006                                    ;       $OCTAL          OCTAL TYPEOUT ROUTINE
2007
2008                                    ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
2009                                    ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
2010                                    ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.
2011
2012  014054  012737  170101  014222  BITYP$: MOV     #170101,.PR     ;SET BIT FLAG ANS 16. CHARACTER COUNT
2013  014062  000411                  BR      .PTIT           ;NOW TYPE IT IN BIT FORM
2014  014064  112737  000001  014222  PRINTR: MOVB    #1,.PR          ;SET ZERO FILL SWITCH
2015  014072  000402                  BR      .+6             ;SKIP
2016  014074  005037  014222          PRINTS: CLR     .PR             ;SUPRESS LEADING ZERO'S
2017  014100  112737  177772  014223          MOVB    #-6,.PR+1       ;SET COUNT
2018  014106  010446                  .PTIT:  MOV     R4,-(6)         ;SAVE R4
2019  014110  012704  014224                  MOV     #.PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR.
2020  014114  105014                  CLRB    (4)             ;CLEAR FIRST BYTE
2021  014116  000411                  BR      .PRF            ;ROTATE FIRST BIT
2022  014120  105014                  .PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
2023  014122  032737  000100  014222          BIT     #100,.PR        ;BIT TYPING MODE?
2024  014130  001004                  BNE     .PRF            ;YES - SKIP 2 ROTATES
2025  014132  006105                  ROL     TTY             ;ROTATE BIT INTO C
2026  014134  106114                  ROLB    (4)             ;PACK IT
2027  014136  006105                  ROL     TTY             ;ROTATE BIT INTO C
2028  014140  106114                  ROLB    (4)             ;PACK IT
2029  014142  006105                  .PRF:   ROL     TTY             ;ROTATE BIT INTO C
2030  014144  106114                  ROLB    (4)             ;PACK IT
2031  014146  105714                  TSTB    (4)             ;IS IT ZERO?
2032  014150  001402                  BEQ     .+6             ;SKIP INC
2033  014152  105237  014222          INCB    .PR             ;SET FILL SWITCH
2034  014156  105737  014222          TSTB    .PR             ;CHECK FILL SWITCH
2035  014162  001402                  BEQ     .+6             ;SKIP BITSET
2036  014164  152724  000060          BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
2037  014170  105237  014223          INCB    .PR+1           ;INC COUNT
2038  014174  001351                  BNE     .PRL            ;REPEAT
2039  014176  022704  014224          CMP     #.PR+2,R4       ;EMPTY BUFFER?
2040  014202  001002                  BNE     .+6             ;SKIP IF NOT
2041  014204  112724  000060          MOVB    #'0,(4)+        ;LOAD 1 ZERO
2042  014210  105014                  CLRB    (4)             ;NULL TERMINATOR
2043  014212  000004  014224          TYPE    .PR+2           ;TYPE IT
2044  014216  012604                  MOV     (6)+,R4         ;RESTORE R4
2045  014220  000207                  RTS     PC              ;RETURN
2046  014222  000012          .PR:    .BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
2047  014246  012777  014374  000126  PDOWN$: MOV     #ILLUP,@PUVEC$  ;SET FOR FAST UP
2048  014254  012777  000340  000122          MOV     #340,@PUVEC$+2  ;PRIO:7
2049  014262  010046                          MOV     R0,-(6)         ;PUSH R0 ON STACK
2050  014264  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
2051  014266  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
2052  014270  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
2053  014272  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
2054  014274  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
2055  014276  010637  014400                  MOV     SP,.SAVR6       ;SAVE SP
2056  014302  012777  014312  000072          MOV     #PUP$,@PUVEC$   ;SET UP VECTOR
2057  014310  000000                          HALT                    ;WAIT FOR PF
2058
2059  014312  013706  014400          PUP$:   MOV     .SAVR6,SP       ;GET SP
2060  014316  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
2061  014320  005201                  1$:     INC     R1              ;WAIT FOR THE INC
2062  014322  001376                          BNE     1$              ;OF  WORD
2063  014324  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
2064  014326  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
2065  014330  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
2066  014332  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
2067  014334  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
2068  014336  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
2069  014340  012737  014246  000024          MOV     #PDOWN$,@#24    ;SET UP THE POWER DOWN VECTOR
2070  014346  012737  000340  000026          MOV     #340,@#26       ;PRIO:7
2071  014354  000004  014360                  TYPE    .+2             ;.ASCIZ <15><12>"POWER"
2072  014370  000137  006322                  JMP     RESTA.          ;JMP TO USER ADDRESS
2073
2074  014374  000000          ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
2075  014376  000776                  BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
2076
2077  014400  000000          .SAVR6: 0                       ;PUT THE SP HERE
2078  014402  000024  000026  PUVEC$: 24,26                   ;POWER UP VECTOR
2079
2080
2081  014406  000002          YESRT:  RTI                     ;RETURN FROM TRACE TRAP
```

```
                                ;*****************************************************************
                                ;IOT HANDLER - REENTERENT ROUTINE TO INDICATE A FALSE
                                ;              INTERRUPT OR TRAP, OR TO TYPE A MESSAGE
                                ;*****************************************************************

014410  022716  001000          IOTRAP: CMP     #1000,  (SP)    ;CHECK RETURN ADDRESS
014414  002440                          BLT     IOT$            ;BRANCH IF TYPE COMMAND
014416  162716  000004                  SUB     #4,     (SP)    ;GET VECTOR ADDRESS
014422  000004  014426                  TYPE,   .+2             ;TYPE MESSAGE
014426  005015  047125  054105          .ASCIZ  <15><12>"UNEXPECTED INTERUPT TO "
014434  042520  052103  042105
014442  044440  052116  051105
014450  050125  020124  047524
014456  000040
014460  012605                          MOV     (SP)+,TTY       ;TYPE (SP)+ IN OCTAL
014462  004737  014074                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
014466  005726                          TST     (SP)+           ;POP STACK
014470  000004  014474                  TYPE,   .+2             ;TYPE MESSAGE
014474  043040  047522  020115          .ASCIZ  " FROM "
014502      000
        014504                          .EVEN
014504  011605                          MOV     (SP),TTY        ;TYPE (SP) IN OCTAL
014506  004737  014074                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
014512  000000                          HALT                    ;FATAL ERROR
014514  000002                          RTI                     ;CONTINUE IF DESIRED

                                ;       $TYPE           MESSAGE TYPEOUT ROUTINE

                                ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
                                ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
                                ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
                                ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
                                ;THE MESSAGE WHICH IS INLINE ASCII.

014516  010546                  IOT$:   MOV     TTY,-(6)        ;SAVE TTY
014520  017605  000002                  MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
014524  032705  177400                  BIT     #177400,TTY     ;IS IT A TYPEM?
014530  001004                          BNE     1$              ;NO
014532  010537  014622                  MOV     TTY,.TYPE       ;GET THE CHARACTER
014536  012705  014622                  MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
014542  105715                  1$:     TSTB    (TTY)           ;TERMINATOR?
014544  001406                          BEQ     2$              ;GET OUT IF SO
014546  112537  177566                  MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
014552  105737  1775[illegible]          TSTB    @#177564        ;IS THE PRINTER READY
014556  100375                          BPL     .-4             ;WAIT UNTIL IT IS
014560  000770                          BR      1$              ;GET THE NEXT CHARACTER
014562  017646  000002          2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
014566  062766  000002  000004          ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
014574  022666  000002                  CMP     (6)+,2(6)       ;IS IT .+2?
014600  001006                          BNE     3$              ;NO
014602  062705  000002                  ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
014606  042705  000001                  BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
014612  010566  000002                  MOV     TTY,2(6)        ;RESTORE ADDRESS
014616  012605                  3$:     MOV     (6)+,TTY        ;RESTORE TTY
014620  000002                          RTI                     ;RETURN
```

```
2138  014622  000000                    .TYPE:  0                       ;CHARACTER TYPE LOCATION
2139
2140  014624  022737  000176  001210    TLU:    CMP     #SWREG,SWR
2141  014632  001023                            BNE     1$
2142  014634  000004  014716                    TYPE    ,SWREQ
2143  014640  013705  000176                    MOV     SWREG,TTY       ;TYPE SWREG IN OCTAL
2144  014644  004737  014064                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
2145  014650  000004  014706                    TYPE    ,NEWIS
2146  014654  004537  013520                    JSR     R5,READIN
2147  014660  015054                            .WORD   TMP1
2148  014662  001360                            BNE     CNTLU
2149  014664  022737  000020  014704           CMP     #20,CNT
2150  014672  001403                            BEQ     1$
2151  014674  013777  015054  164306           MOV     TMP1,@SWR
2152  014702  000207                    1$:     RTS     PC
2153
2154  014704  000000                    CNT:    0
2155
2156  014706  020040  042516  036527    NEWIS:  .ASCIZ  "  NEW= "
2157  014714  000040
2158  014716  005015  053523  036522    SWREG:  .ASCIZ  <15><12>"SWR= "
2159  014724  000040
2160
2161
2162  014726  013746  000006            SUSWRR: MOV     6,-(SP)
2163  014732  013746  000004                    MOV     4,-(SP)
2164  014736  012737  014756  000004           MOV     #1$,4
2165  014744  022777  177777  164236           CMP     #-1,@SWR
2166  014752  001402                            BEQ     2$
2167  014754  000407                            BR      3$
2168  014756  022626                    1$:     CMP     (SP)+,(SP)+
2169  014760  012737  000176  001210    2$:     MOV     #SWREG,SWR
2170  014766  012737  000174  001212            MOV     #DISPREG,DISPLAY
2171  014774  012637  000004            3$:     MOV     (SP)+,4
2172  015000  012637  000006                    MOV     (SP)+,6
2173  015004  000207                            RTS     PC
2174
2175
2176  015006  022737  000176  001210    KBDINT: CMP     #SWREG,SWR
2177  015014  001016                            BNE     1$
2178  015016  005037  015054                    CLR     TMP1
2179  015022  113737  177562  015054           MOVB    177562,TMP1
2180  015030  142737  000200  015054           BICB    #200,TMP1
2181  015036  122737  000007  015054           CMPB    #7,TMP1
2182  015044  001002                            BNE     1$
2183  015046  004737  014624                    JSR     PC,CNTLU
2184  015052  000207                    1$:     RTS     PC
2185
2186  015054  000000                    TMP1:   0
2187
2188
2189  015056  005015  177777  000377    RETURN: .ASCIZ  <15><12><377><377><377>
2190  015064  020040     000            SPACE:  .ASCIZ  "  "
2191  015067     015  177412  040515    MSGMDN: .ASCIZ  <15><12><377>"MAINDEC-11-DZDJB-D   DJ11 EXERCISER"
2192  015074  047111  042504  026503
2193  015102  030461  042055  042132
```

```
2194  015110  041112  042055  020040
2195  015116  042040  030512  020061
2196  015124  054105  051105  044503
2197  015132  042523  000122
2198  015136  005015  044506  051522  MSGADR: .ASCIZ  <15><12>"FIRST DJ11 ADDRESS:  "
2199  015144  020124  045104  030461
2200  015152  040440  042104  042522
2201  015160  051523  020072  000040
2202  015166  005015  042526  052103  MSGVEC: .ASCIZ  <15><12>"VECTOR ADDRESS:  "
2203  015174  051117  040440  042104
2204  015202  042522  051523  020072
2205  015210  000040
2206  015212  005015  047516  020056  MSGNUM: .ASCIZ  <15><12>"NO. OF DJ11'S:  "
2207  015220  043117  042040  030512
2208  015226  023461  035123  020040
2209  015234      000
2210  015235      015  050012  047522  MSGPRG: .ASCIZ  <15><12>"PROGRAM #:  "
2211  015242  051107  046501  021440
2212  015250  020072  000040
2213  015254  005015  047516  042040  MSG01:  .ASCIZ  <15><12>"NO DJ11'S!"
2214  015262  030513  023461  020523
2215  015270      000
2216  015271      015  050012  047522  MSGP2:  .ASCIZ  <15><12>"PROG2:  CONTINUOUS ECHO EXERCISER"<15><12>
2217  015276  031107  020072  041440
2218  015304  047117  044524  052516
2219  015312  052517  020123  041505
2220  015320  047510  042440  042530
2221  015326  041522  051511  051105
2222  015334  005015      000
2223  015337      015  025012  041505  MSGP3:  .ASCIZ  <15><12>"*ECHO TEST*"<15><12>
2224  015344  047510  052040  051505
2225  015352  025124  005015      000
2226          015360                  .EVEN
2227  015360  000000                  END:    0
2228          000001                  .END
```

```
ADRNIT  007714            822     823     1020    1021    1045#
BEGIN   005306            493     552#
BELL  = 000007            459#    1882    1907
BINARY  007760           1045     1061#
BITYP$  014054           2012#
BIT0  = 000001            460#
BIT1  = 000002            461#
BIT10 = 002000            470#
BIT11 = 004000            471#
BIT12 = 010000            472#
BIT13 = 020000            473#    643
BIT14 = 040000            474#
BIT15 = 100000            475#
BIT2  = 000004            462#    800
BIT3  = 000010            463#
BIT4  = 000020            464#    667     807
BIT5  = 000040            465#
BIT6  = 000100            466#
BIT7  = 000200            467#
BIT8  = 000400            468#    724
BIT9  = 001000            469#    798
BUFMSK  001266            527#    1602*   1603*   1605    1606    1777    1853    1857
BUFNIT  011742           1599#    1601
BUFSIZ  001264            526#    1597    1599*   1602    1644    1779    1855
CNT     014704           1947*    1965*   2149    2154#
CNTLU   014624            662     2140#   2148    2183
CNTNIT  007716            824     825     1022    1023    1046#
CNTTAB= 004307            550#    826*    842     849     1009    1015*   1495    1499*   1569*
DEVADR  001272            530#    608     647     664     683     784     1359    1615    1677
DISPLA  001212            504#    2170*
DISPRE  000174            489#    2170
DJMAP   005466            584#
DONE    013242            600     862     1876#
EMT$    013326            561     1904#
END     015360           1051     1053    1055    1057    1059    1604    2227#
ERRORS  001202            499#    570*    1908*   1925
ERROR$  013460           1918     1928#
FLTADR  001276            532#    584
FLTVEC  001300            533#
FORGND  006660            842#
FORP2   010700           1401#    1407
FORP3   012204           1669#
GETLEN  005764            644     660#
GETVEC  005550            597     606#
HLT   = 104000            447#    984     1001    1537    1552    1814    1828    1833    1839
HLTADR  013456           1912*    1913*   1914    1915    1926#
HLTCT$  013454           1914*    1925#   1929*
ICNT    001200            498#    552     764     1339    1585
ILLUP   014374           2047     2074#
INHRE   014012           1940*    1942*   1996    2004#
INPUT   014014           1948     1978    1979    2005#
IOTRAP  014410            557     2088#
IOT$    014516           2089     2117#
KBDINT  015006           1877     1904    1922    2176#
LEVEL7= 000340            476#
MARK    001262            525#    814*    820     828*    1382*   1384    1389*   1508*   1510*   1513    1560*   1562*   1565
```

```
                    1636*   1640    1655*
MASK    004306       549#    550     684     997    1550    1826
MEMX3   012254      1684#   1690
MSGADR  015136       646    2198#
MSGMDN  015067       645    2191#
MSGNUM  015212       652    2206#
MSGPRG  015235       715    2210#
MSGP2   015271      1386    2216#
MSGP3   015337      1645    2223#
MSGVEC  015166       649    2202#
MSGD1   015254       596    2213#
N       000040       385#   1062#   1063#   1064#   1065#   1066#   1067#   1068#   1069#   1070#   1071#   1072#   1073#
                    1074#   1075#   1076#   1077#   1078#   1079#   1080#   1081#   1082#   1083#   1084#   1085#   1086#
                    1087#   1088#   1089#   1090#   1091#   1092#   1093#   1094#   1095#   1096#   1097#   1098#   1099#
                    1100#   1101#   1102#   1103#   1104#   1105#   1106#   1107#   1108#   1109#   1110#   1111#   1112#
                    1113#   1114#   1115#   1116#   1117#   1118#   1119#   1120#   1121#   1122#   1123#   1124#   1125#
                    1126#   1127#   1128#   1129#   1130#   1131#   1132#   1133#   1134#   1135#   1136#   1137#   1138#
                    1139#   1140#   1141#   1142#    143#   1144#   1145#   1146#   1147#   1148#   1149#   1150#   1151#
                    1152#   1153#   1154#   1155#    156#   1157#   1158#   1159#   1160#   1161#   1162#   1163#   1164#
                    1165#   1166#   1167#   1168#   1169#   1170#   1171#   1172#   1173#   1174#   1175#   1176#   1177#
                    1178#   1179#   1180#   1181#   1182#   1183#   1184#   1185#   1186#   1187#   1188#   1189#   1190#
                    1191#   1192#   1193#   1194#   1195#   1196#   1197#   1198#   1199#   1200#   1201#   1202#   1203#
                    1204#   1205#   1206#   1207#   1208#   1209#   1210#   1211#   1212#   1213#   1214#   1215#   1216#
                    1217#   1218#   1219#   1220#   1221#   1222#   1223#   1224#   1225#   1226#   1227#   1228#   1229#
                    1230#   1231#   1232#   1233#   1234#   1235#   1236#   1237#   1238#   1239#   1240#   1241#   1242#
                    1243#   1244#   1245#   1246#   1247#   1248#   1249#   1250#   1251#   1252#   1253#   1254#   1255#
                    1256#   1257#   1258#   1259#   1260#   1261#   1262#   1263#   1264#   1265#   1266#   1267#   1268#
                    1269#   1270#   1271#   1272#   1273#   1274#   1275#   1276#   1277#   1278#   1279#   1280#   1281#
                    1282#   1283#   1284#   1285#   1286#   1287#   1288#   1289#   1290#   1291#   1292#   1293#   1294#
                    1295#   1296#   1297#   1298#   1299#   1300#   1301#   1302#   1303#   1304#   1305#   1306#   1307#
                    1308#   1309#   1310#   1311#   1312#   1313#   1314#   1315#   1316#   1317#
NEWIS   014706      2145    2156#
OPEN    000000       477#    509     510     511     512     513     514     515     516     517     518     519     520
                     521     522     523     524
PC      000007       458#    553#    648#    651#    654#    662#   1877#   1904#   1916#   1918#   1922#   1936#   1941#
                    2002#   2045#   2098#   2105#   2144#   2152#   2173#   2183#   2184#
PCNT    001204       500#    571#    572#   1878#   1879#   1894
PDOWN$  014246       559    2047#   2069
PHRASE  010360      1047    1318#
PRINTR  014064       648    1916    1933    2014#   2144
PRINTS  014074       651     654    2016#   2098    2105
PROGAD  006344       745     748#
PROGNO  006344       717     740     747#
PROG1   006354       749     763#
PROG2   010462       750    1338#
PROG3   011666       751    1584#   1847
PS      177776       449#    765#    834#   1340#   1395#   1586#   1663#   1968    1972#
PSTEMP  013664      1968#   1972    1975#
PUP$    014312      2056    2059#
PUVEC$  014402      2047#   2048#   2056#   2078#
P1INIT  005422       783#
P2INIT  010530      1358#
P2RISR  011512      1413    1417    1421    1425    1429    1433    1437    1441    1445    1449    1453    1457    1461
                    1465    1469    1473    1528#
P2XISR  011320      1415    1419    1423    1427    1431    1435    1439    1443    1447    1451    1455    1459    1463
                    1467    1471    1475    1482#
```

```
P3INIT  012000          1614*
P3RISR  013024          1696    1700    1704    1708    1712    1716    1720    1724    1728    1732    1736    1740    1744
                        1748    1752    1756    1804*
P3XISR  012674          1698    1702    1706    1710    1714    1718    1722    1726    1730    1734    1738    1742    1746
                        1750    1754    1758    1765*
RCVCNT= 003307           547*    825*    855    1005*   1023*
RCVISR  007472           868     872     876     880     884     888     892     896     900     904     908     912     916
                         920     924     928     974*
RCVLVL  001302           534*    788    1363    1619
RCVTAB  002306           543*    823*    996    1007*   1021*   1501    1504    1555    1557*   1568*   1643*   1831    1837
                        1851*   1852*   1853    1855*   1857*   1858    1859*
READIN  013520           716     730    1940*   2146
READ$   013566          1941    1977*
REPORT  005704           632     643*
RESTAR  006322           495     714     725     740*    1892    2072
RETURN  015056          1515    1570    2189*
RISR0   006756           786     868*
RISR1   006776           872*
RISR10  007156           900*
RISR11  007176           904*
RISR12  007216           908*
RISR13  007236           912*
RISR14  007256           916*
RISR15  007276           920*
RISR16  007316           924*
RISR17  007336           928*
RISR2   007016           876*
RISR3   007036           880*
RISR4   007056           884*
RISR5   007076           888*
RISR6   007116           892*
RISR7   007136           896*
R0      =%000000         450*    554*    555*    556*    557*    558*    559*    560*    561*    562*    564*    573*    574
                         575*    576*    577     584*    589     591*    663*    680*    682*    703*    719*    722*    726*
                         733*    734     740*    742     744*    745     783*    811*    812     813*    831*    832     868*
                         870*    872*    874*    876*    878*    880*    882*    884*    886*    888*    890*    892*    894*
                         896*    898*    900*    902*    904*    906*    908*    910*    912*    914*    916*    918*    920*
                         922*    924*    926*    928*    930*    940     945     954     967*    979     995    1024    1038*
                        1358*   1379*   1380    1381*   1392*   1393    1401*   1403*   1405    1413*   1415*   1417*   1419*
                        1421*   1423*   1425*   1427*   1429*   1431*   1433*   1435*   1437*   1439*   1441*   1443*   1445*
                        1447*   1449*   1451*   1453*   1455*   1457*   1459*   1461*   1463*   1465*   1467*   1469*   1471*
                        1473*   1475*   1485    1490    1506    1521*   1532    1549    1558    1577*   1597*   1598*   1599
                        1600*   1614*   1633*   1634    1635*   1660*   1661    1676*   1682*   1684*   1686*   1688    1696*
                        1698*   1700*   1702*   1704*   1706*   1708*   1710*   1712*   1714*   1716*   1718*   1720*   1722*
                        1724*   1726*   1728*   1730*   1732*   1734*   1736*   1738*   1740*   1742*   1744*   1746*   1748*
                        1750*   1752*   1754*   1756*   1758*   1768    1773    1784    1797*   1809    1825    1860    1874*
                        1884*   2049    2068*
R1      =%000001         451*    587*    592*    596     608*    609*    612*    628*    629*    633*    636*    637*    664*
                         666*    667     669*    670     672*    678*    679*    683*    686     701*    720*    721*    727*
                         732*    766*    768*    772*    773*    784*    790     796     797*    800*    801*    807     809*
                         810     812*    820     830     842*    844     846*    849*    851*    852*    855*    957     959*
                         938     940*    941     943     961*    966*    975     979*    980    1031*   1037*   1341*   1343*
                        1347*   1348*   1359*   1365    1371    1372*   1378    1380*   1384    1391    1493    1485*   1486
                        1488    1493*   1513*   1520*   1529    1532*   1533    1565*   1576*   1587*   1589*   1593*   1594*
                        1615*   1621    1627    1628*   1632    1634*   1640    1659    1669*   1671    1673*   1677*   1678*
                        1681*   1766    1768*   1769    1771    1774*   1791*   1796*   1805    1809*   1810    1867*   1873*
```

R2	=%000002	1943 1946* 1951* 1958* 1960* 1962* 1964* 1971* 2050 2060* 2061* 2067* 452* 585* 616* 623 630* 634* 635* 665* 669 673* 674* 675* 676* 684* 692 702* 767* 769* 771* 774* 785* 787* 788* 793* 794* 843* 847* 850* 853* 856* 860* 939 943* 944* 945* 954* 955* 959* 965* 976 980* 982 990* 991 998 1036* 1342* 1344* 1346* 1349* 1360* 1362* 1363* 1368* 1369* 1484 1488* 1489* 1490* 1502 1506* 1507* 1511* 1519* 1530 1533* 1535 1545 1550 1557 1568 1575* 1588* 1590* 1592* 1595* 1616* 1618* 1619* 1624* 1625* 1638 1644* 1648* 1652* 1658* 1670* 1674* 1767 1771* 1772* 1773* 1784* 1785* 1789* 1795* 1806 1810* 1812 1821 1826 1851 1859 1872* 1944 1948* 1949 1952 1970* 2051 2066*
R2SR0	010720	1361 1413*
R2SR1	010740	1417*
R2SR10	011120	1445*
R2SR11	011140	1449*
R2SR12	011160	1453*
R2SR13	011200	1457*
R2SR14	011220	1461*
R2SR15	011240	1465*
R2SR16	011260	1469*
R2SR17	011300	1473*
R2SR2	010760	1421*
R2SR3	011000	1425*
R2SR4	011020	1429*
R2SR5	011040	1433*
R2SR6	011060	1437*
R2SR7	011100	1441*
R3	=%000003	453* 624* 631* 685* 699* 786* 787 789 790* 791* 792 793 795 796* 819 952 956* 958* 961 962* 977 996* 997* 998 1000* 1016* 1017* 1018* 1019* 1026* 1028* 1031 1035* 1361* 1362 1364 1365* 1366* 1367 1368 1370 1371* 1383 1531 1545* 1546* 1547* 1548* 1549* 1558* 1559* 1563* 1574* 1617* 1618 1620 1621* 1622* 1623 1624 1626 1627* 1637 1639 1645* 1646 1657* 1782 1786* 1788* 1791 1792* 1807 1862* 1864* 1867 1871* 1945 1952* 1953 1955* 1956 1964 1969* 1977 1978* 1979 2001* 2052 2065*
R3SR0	012274	1617 1696*
R3SR1	012314	1700*
R3SR10	012474	1728*
R3SR11	012514	1732*
R3SR12	012534	1736*
R3SR13	012554	1740*
R3SR14	012574	1744*
R3SR15	012614	1748*
R3SR16	012634	1752*
R3SR17	012654	1756*
R3SR2	012334	1704*
R3SR3	012354	1708*
R3SR4	012374	1712*
R3SR5	012414	1716*
R3SR6	012434	1720*
R3SR7	012454	1724*
R4	=%000004	454* 686* 688 693* 694* 695 696 697 698 819* 827 978 991* 992* 993* 994* 995* 996 1024* 1025* 1029* 1034* 1383* 1388 1637* 1654 1808 1821* 1822* 1823* 1824* 1825* 1860* 1861* 1865* 1870* 2019 2019* 2039 2044* 2053 2064*
R5	=%000005	455* 688* 689* 690* 691* 692* 695* 696* 697* 698* 716* 730* 915* 816* 817* 818* 1604* 1605* 1606* 1642 1643 1646* 1651* 1946 1973* 2054 2063* 2146*

```
SAVIT   001214          506*    731     732
SCOPE = 104400          446*
SELPRO  006172          713*    718     741     743
SIXBIT  007752         1049    1060*
SP    =%000006          457*    552*    606*    616     617*    618     619     622     764*    952*    962     1339*   1585*
                       1782*   1792    2055    2059*   2088    2090*   2097    2099    2104    2162*   2163*   2168    2171
                       2172
SPACE   015064         1934    2190*
SUSWRR  014726          553    2162*
SVSW0   001222          509*    720     727     812     1380    1634
SVSW1   001224          510*
SVSW10  001242          517*
SVSW11  001244          518*
SVSW12  001246          519*
SVSW13  001250          520*
SVSW14  001252          521*
SVSW15  001254          522*
SVSW16  001256          523*
SVSW17  001260          524*
SVSW2   001226          511*
SVSW3   001230          512*
SVSW4   001232          513*
SVSW5   001234          514*
SVSW6   001236          515*
SVSW7   001240          516*
SWITCH  006252          728*    735
SWR     001210          503*    643     660     724     798     815     1016    1880    1905    1909    1919    2140    2151*
                       2165    2169*   2176
SWREG   000176          490*    660     2140    2143    2169    2176
SWREQ   014716         2142    2158*
SWRSAV  001220          508*
SW10  = 002000          397*   1880    1904    1905
SW11  = 004000          396*
SW12  = 010000          395*   1881    1884
SW13  = 020000          394*   1909
SW14  = 040000          393*
SW15  = 100000          392*
SW8   = 000400          400*
SW9   = 001000          399*   1923
TIMES   001216          507*    826
TMP1    015054         2147    2151    2178*   2179*   2180*   2181    2186*
TTY   =%000005          456*    647*    650*    653*   1915*   1932*   2025*   2027*   2029*   2097*   2104*   2117    2119*
                       2119    2121    2122*   2123    2125    2133*   2134*   2135    2136*   2143*
TYPE  = 000004          448*    598     645     646     647     652     715     729    1882    1883    1907    1911    1917
                       1934    1990    1993    2000    2043    2071    2091    2100    2142    2145
UNITS   001270          529*    596*    624     653     663     682     719     734     832    1393    1661    1678
VECADR  001274          531*    618*    650     785    1360    1616
XISR0   006766          870*
XISR1   007006          874*
XISR10  007166          902*
XISR11  007206          906*
XISR12  007226          910*
XISR13  007246          914*
XISR14  007266          918*
XISR15  007306          922*
XISR16  007326          926*
```

DZDJBD.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
XISR17  007346          930#
XISR2   007026          878#
XISR3   007046          882#
XISR4   007066          886#
XISR5   007106          890#
XISR6   007126          894#
XISR7   007146          898#
XMTCNT  003306          546#    547     824*    946     949*    1022*   1387*   1491    1494*   1503*   1516*   1571*
XMTISR  007356          870     874     878     882     886     890     894     898     902     906     910     914     918
                        922     926     930     937#
XMTLVL  001304          535#    794     1369    1625
XMTTAB  001306          541#    766     822*    948     95      953*    1020*   1341    1386*   1493    1497*   1501*   1502*
                       1515*   1570*   1587    1642*   1669    1774    1775*   1776*   1777    1779*   1780    1783*   1849
                       1858#
X2SR0   010730          1415#
X2SR1   010750          1419#
X2SR10  011130          1447#
X2SR11  011150          1451#
X2SR12  011170          1455#
X2SR13  011210          1459#
X2SR14  011230          1463#
X2SR15  011250          1467#
X2SR16  011270          1471#
X2SR17  011310          1475#
X2SR2   010770          1423#
X2SR3   011010          1427#
X2SR4   011030          1431#
X2SR5   011050          1435#
X2SR6   011070          1439#
X2SR7   011110          1443#
X3SR0   012304          1698#
X3SR1   012324          1702#
X3SR10  012504          1730#
X3SR11  012524          1734#
X3SR12  012544          1738#
X3SR13  012564          1742#
X3SR14  012604          1746#
X3SR15  012624          1750#
X3SR16  012644          1754#
X3SR17  012664          1758#
X3SR2   012344          1706#
X3SR3   012364          1710#
X3SR4   012404          1714#
X3SR5   012424          1718#
X3SR6   012444          1722#
X3SR7   012464          1726#
YESRT   014406           555     565*    2081#
$ENDAD= 013310           486    1887#
.       = 015362           478#    482     485#    488#    492#    494#    497#    541#    543#    546#    549#    729     730
                        1329#   1887    1894    1911    1912#   1917    1918#   1920    1925    1926    1982    1990    2005#
                        2015    2032    2035    2040    2046#   2071    2075    2091    2100    2103#   2127    2226#
.BIT    = 122000           385#    392     393     394     395     396     397     398     399     400     1880    1881    1884
                        1904    1909    1923
.PR     014222          2012*   2014*   2016*   2017*   2019    2023    2033*   2034    2037*   2039    2043    2046#
.PRF    014142          2021    2024    2029#
.PRL    014120          2022#   2038
```

```
.PTIT   014106          2013    2018#
.SAVR6  014400          2055*   2059    2077#
.TBIT   013324          1894#
.TYPE   014622          1992*   1993    2121*   2122    2138#
```

```
BCP     1062* 1063  1064  1065  1066  1067  1068  1069  1070  1071  1072  1073  1074  1075  1076
        1077  1078  1079  1080  1081  1082  1083  1084  1085  1086  1087  1088  1089  1090  1091
        1092  1093  1094  1095  1096  1097  1098  1099  1100  1101  1102  1103  1104  1105  1106
        1107  1108  1109  1110  1111  1112  1113  1114  1115  1116  1117  1118  1119  1120  1121
        1122  1123  1124  1125  1126  1127  1128  1129  1130  1131  1132  1133  1134  1135  1136
        1137  1138  1139  1140  1141  1142  1143  1144  1145  1146  1147  1148  1149  1150  1151
        1152  1153  1154  1155  1156  1157  1158  1159  1160  1161  1162  1163  1164  1165  1166
        1167  1168  1169  1170  1171  1172  1173  1174  1175  1176  1177  1178  1179  1180  1181
        1182  1183  1184  1185  1186  1187  1188  1189  1190  1191  1192  1193  1194  1195  1196
        1197  1198  1199  1200  1201  1202  1203  1204  1205  1206  1207  1208  1209  1210  1211
        1212  1213  1214  1215  1216  1217  1218  1219  1220  1221  1222  1223  1224  1225  1226
        1227  1228  1229  1230  1231  1232  1233  1234  1235  1236  1237  1238  1239  1240  1241
        1242  1243  1244  1245  1246  1247  1248  1249  1250  1251  1252  1253  1254  1255  1256
        1257  1258  1259  1260  1261  1262  1263  1264  1265  1266  1267  1268  1269  1270  1271
        1272  1273  1274  1275  1276  1277  1278  1279  1280  1281  1282  1283  1284  1285  1286
        1287  1288  1289  1290  1291  1292  1293  1294  1295  1296  1297  1298  1299  1300  1301
        1302  1303  1304  1305  1306  1307  1308  1309  1310  1311  1312  1313  1314  1315  1316
BITYPE     1#   476#
BLKBLK     1#
BUFHDR     1#
DUMP       1#   476#  647   1915  2143
DUMP18     1#
FIL3LK     1#
LNKBLK     1#
ODT11      1#
ODT11X     1#
POP        1#   476#  964   1033  1518  1573  1657  1794  1869  1969  2063
PRINT      1#   476#  728   1911  1917  1989  2071
PUSH       1#   476#  937   974   1482  1528  1638  1765  1804  1943  2049
SCOPE.     1#
SCOP.      1#
SDUMP      1#   476#  650   653   2097  2104
TRACE      1#
TRNBLK     1#
TYPEM      1#
WRITE      1#
$CATCH     1#
$CMTAG     1#
$CNTL      1#   2139
$END       1#   1876
$EQUAT     1#   446
$HLT       1#   1896
$IOCAT   385#   478
$IOT     385#
$KBIN      1#   2175
$KRAT      1#
$LOADR     1#
$OCTAL     1#   2006
$POWER     1#   2047
$RAND      1#
$RAND4     1#
$READ      1#   1977
$SCOPE     1#
$SETUP     1#
$SRAT      1#
$SWDOC     1#   389
```

$SWRDF	1#	502
$SWRRR	1#	2161
$TRAP	1#	
$TYPE	1#	2109
$URAT	1#	
.SCOP	1#	
.SCOPE	1#	

```
ADC     1403  1686  1879
ADD      591   633   634   679   692   701   702   791   846   859   945   995  1366  1490  1502
        1549  1605  1622  1673  1681  1773  1925  1878  1931  2130  2133
ASL      673   674   675   676   691   744   811   817   818   828   944  1018  1019  1379  1389
        1409  1633  1655  1772  1958  1960  1962
ASLB     994  1548  1824
ASR      813   955  1025  1381  1507  1559  1598  1600  1635  1785  1861
BCC      677   829  1390  1656
BCS      590  1959  1961  1963
BEQ      632   725   741   821   947   983   999  1006  1011  1385  1492  1500  1536  1551  1641
        1647  1813  1827  1838  1850  1885  1906  1950  1954  1980  1986  1997  2032  2035  2124
        2150  2166
BGT      578
BIC      690   694   816   834   961   990   997  1000    10  1395  1513  1599  1606  1663  1791
        1857  1928  2134
BICB    1984  2180
BIS      765   801   809  1031  1340  1565  1586  1678  1867  1964
BISB    2036
BIT      643   667   724   798   807   820   982  1384  1535  1640  1777  1812  1853  1880  1905
        1909  1956  2023  2119
BITB    1550  1826
BLO      743
BLT     2089
BMI     1010  1930
BNE      597   644   661   668   681   700   704   714   718   723   735   770   775   799   808
         833   845   848   854   858   861  1345  1350  1394  1404  1406  1496  1505  1556  1591
        1596  1601  1649  1653  1662  1672  1675  1683  1687  1689  1778  1781  1832  1854  1881
        1910  1957  1988  1995  2024  2038  2040  2062  2120  2132  2141  2148  2177  2182
BPL      671   687   942   960   981  1030  1487  1512  1534  1564  1770  1790  1811  1866  1920
        1982  2127
BR       593   614   639   951   963  1008  1032  1407  1498  1514  1517  1543  1566  1572  1650
        1690  1793  1820  1836  1856  1868  1935  1966  1991  1998  2013  2015  2021  2075  2128
        2167
CLC     1402  1685
CLR      568   570   571   572   587   595   628   629   678   726   768   783   797   953   956
        1026  1343  1358  1508  1560  1589  1614  1783  1786  1862  1942  1951  2016  2060  2178
CLRB     773   992  1348  1546  1594  1651  1775  1822  1999  2020  2022  2042
CLZ     1967
CMP      577   619   660   734   742   789   795   810   830   832  1364  1370  1391  1393  1620
        1626  1659  1661  1979  2039  2088  2131  2140  2149  2165  2168  2176
CMPB     998  1949  1953  1985  1987  1994  2181
COMB     693  1499  156.
DEC      631   680   699   703   722   769   774   847   853   860   959  1029  1344  1349  1511
        1563  1590  1595  1603  1648  1652  1674  1682  1789  1865  1965
DECB     949  1005  1015  1494  1929
EMT      447
HALT     482   599  1921  2057  2074  2106
INC      592   733   772   831   852   950  1007  1347  1392  1497  1593  1660  1776  1852  1908
        2061
INCB    2033  2037
IOT      448   482   576   630
JMP      493   495   600   745   862  1847  1892  2072
JSR      553   648   651   654   662   716   730   868   870   872   874   876   878   880   892
         884   886   888   890   892   894   896   898   900   902   904   906   908   910   912
         914   916   918   920   922   924   926   928   930  1413  1415  1417  1419  1421  1423
        1425  1427  1429  1431  1433  1435  1437  1439  1441  1443  1445  1447  1449  1451  1453
```

```
        1455  1457  1459  1461  1463  1465  1467  1469  1471  1473  1475  1696  1698  1700  1702
        1704  1706  1708  1710  1712  1714  1716  1718  1720  1722  1724  1726  1728  1730  1732
        1734  1736  1738  1740  1742  1744  1746  1748  1750  1752  1754  1756  1758  1877  1888
        1904  1916  1918  1922  1933  1941  2098  2105  2144  2146  2183
MOV      552   554   555   556   557   558   559   560   561   562   563   565   566   567   569
         573   575   576   584   585   586   596   606   607   608   609   612   616   618   622
         624   630   635   636   637   647   650   653   663   664   665   666   669   672   682
         683   684   685   686   688   695   696   697   698   719   720   721   727   732   740
         764   766   767   771   784   785   786   787   788   790   793   794   796   800   812
         814   815   819   822   823   842   843   849   850   855   856   938   939   940   952
         962   965   966   967   975   976   977   978   979   980   991  1016  1020  1021  1034
        1035  1036  1037  1038  1339  1341  1342  1346  1359  1360  1361  1362  1363  1365  1368
        1369  1371  1372  1380  1382  1383  1386  1387  1401  1483  1484  1485  1501  1515  1519
        1520  1521  1529  1530  1531  1532  1533  1545  1570  1574  1575  1576  1577  1585  1587
        1588  1592  1597  1602  1604  1615  1616  1617  1618  1619  1621  1624  1625  1627  1628
        1634  1636  1637  1638  1639  1642  1643  1644  1645  1657  1658  1669  1670  1676  1677
        1684  1766  1767  1768  1782  1792  1795  1796  1797  1805  1806  1807  1808  1809  1810
        1821  1858  1870  1871  1872  1873  1874  1884  1912  1915  1932  1940  1943  1944  1945
        1946  1947  1948  1968  1969  1970  1971  1972  1977  1978  2001  2012  2018  2019  2044
        2047  2048  2049  2050  2051  2052  2053  2054  2055  2056  2059  2063  2064  2065  2066
        2067  2068  2069  2070  2097  2104  2117  2118  2121  2122  2129  2135  2136  2143  2151
        2162  2163  2164  2169  2170  2171  2172
MOVB     824   825   826   851   943   948   996  1022  1023  1488  1493  1503  1516  1557  1568
        1571  1646  1771  1774  1851  1859  1914  1952  1983  1992  2014  2017  2041  2125  2179
NOP     1889  1890  1891
RESET    763  1338  1584  1886
ROL      958  1028  1510  1562  1788  1864  2025  2027  2029
ROLB    2026  2028  2030
RTI      586   620   635   968  1039  1522  1578  1798  1875  1923  2081  2107  2137
RTS     1936  1973  2002  2045  2152  2173  2184
RTT      565   566
SEC      588   957  1027  1509  1561  1787  1863
SUB      617   954  1024  1506  1558  1779  1784  1855  1860  1913  1955  2090
SWAB     689   993  1547  1823
SXT      564
TRAP     446
TST      574   589   623   670   713   792   827   941  1367  1378  1388  1405  1486  1623  1632
        1654  1671  1688  1769  1831  1849  1919  1996  2099
TSTB     844   857   946  1009  1491  1495  1504  1555  1780  1837  1981  2031  2034  2123  2126
WAIT     613   638
.ASCII  1060  1318
.ASCIZ   730  1319  1912  1918  1991  2072  2092  2101  2156  2158  2189  2190  2191  2198  2202
        2206  2210  2213  2216  2223
.BLKW    541   543   546   549  2005  2046
.BYTE   1061  1062  1063  1064  1065  1066  1067  1068  1069  1070  1071  1072  1073  1074  1075
        1076  1077  1078  1079  1080  1081  1082  1083  1084  1085  1086  1087  1088  1089  1090
        1091  1092  1093  1094  1095  1096  1097  1098  1099  1100  1101  1102  1103  1104  1105
        1106  1107  1108  1109  1110  1111  1112  1113  1114  1115  1116  1117  1118  1119  1120
        1121  1122  1123  1124  1125  1126  1127  1128  1129  1130  1131  1132  1133  1134  1135
        1136  1137  1138  1139  1140  1141  1142  1143  1144  1145  1146  1147  1148  1149  1150
        1151  1152  1153  1154  1155  1156  1157  1158  1159  1160  1161  1162  1163  1164  1165
        1166  1167  1168  1169  1170  1171  1172  1173  1174  1175  1176  1177  1178  1179  1180
        1181  1182  1183  1184  1185  1186  1187  1188  1189  1190  1191  1192  1193  1194  1195
        1196  1197  1198  1199  1200  1201  1202  1203  1204  1205  1206  1207  1208  1209  1210
        1211  1212  1213  1214  1215  1216  1217  1218  1219  1220  1221  1222  1223  1224  1225
        1226  1227  1228  1229  1230  1231  1232  1233  1234  1235  1236  1237  1238  1239  1240
```

```
                1241    1242    1243    1244    1245    1246    1247    1248    1249    1250    1251    1252    1253    1254    1255
                1256    1257    1258    1259    1260    1261    1262    1263    1264    1265    1266    1267    1268    1269    1270
                1271    1272    1273    1274    1275    1276    1277    1278    1279    1280    1281    1282    1283    1284    1285
                1286    1287    1288    1289    1290    1291    1292    1293    1294    1295    1296    1297    1298    1299    1300
                1301    1302    1303    1304    1305    1306    1307    1308    1309    1310    1311    1312    1313    1314    1315
                1316
.ENABL             1     386     387
.END            2228
.ENDC            476    1880    1882    1884    1893    1909    1911    1919    1924    1980    2006    2022    2047    2055    2063
                2072    2073    2123    2138
.EVEN            730    1329    1912    1918    1991    2072    2103    2226
.IF              460    1880    1881    1884    1892    1904    1909    1918    1923    1979    2005    2006    2022    2047    2055
                2063    2071    2072    2119    2129
.IFF            1881    1892    1904    1924    1980    2006    2072    2138
.IIF             392     393     394     395     396     397     398     399     400    1883    1894    1925    1926    2005
.IRP             509     868     938     965     975    1034    1413    1483    1519    1529    1574    1638    1657    1696    1766
                1795    1805    1870    1943    1969    2049    2063
.LIST              1     385     402     476     482     551     579     601     655     705     730     752     835     863     932
                 969    1040    1062    1063    1064    1065    1066    1067    1068    1069    1070    1071    1072    1073    1074
                1075    1076    1077    1078    1079    1080    1081    1082    1083    1084    1085    1086    1087    1088    1089
                1090    1091    1092    1093    1094    1095    1096    1097    1098    1099    1100    1101    1102    1103    1104
                1105    1106    1107    1108    1109    1110    1111    1112    1113    1114    1115    1116    1117    1118    1119
                1120    1121    1122    1123    1124    1125    1126    1127    1128    1129    1130    1131    1132    1133    1134
                1135    1136    1137    1138    1139    1140    1141    1142    1143    1144    1145    1146    1147    1148    1149
                1150    1151    1152    1153    1154    1155    1156    1157    1158    1159    1160    1161    1162    1163    1164
                1165    1166    1167    1168    1169    1170    1171    1172    1173    1174    1175    1176    1177    1178    1179
                1180    1181    1182    1183    1184    1185    1186    1187    1188    1189    1190    1191    1192    1193    1194
                1195    1196    1197    1198    1199    1200    1201    1202    1203    1204    1205    1206    1207    1208    1209
                1210    1211    1212    1213    1214    1215    1216    1217    1218    1219    1220    1221    1222    1223    1224
                1225    1226    1227    1228    1229    1230    1231    1232    1233    1234    1235    1236    1237    1238    1239
                1240    1241    1242    1243    1244    1245    1246    1247    1248    1249    1250    1251    1252    1253    1254
                1255    1256    1257    1258    1259    1260    1261    1262    1263    1264    1265    1266    1267    1268    1269
                1270    1271    1272    1273    1274    1275    1276    1277    1278    1279    1280    1281    1282    1283    1284
                1285    1286    1287    1288    1289    1290    1291    1292    1293    1294    1295    1296    1297    1298    1299
                1300    1301    1302    1303    1304    1305    1306    1307    1308    1309    1310    1311    1312    1313    1314
                1315    1316    1317    1330    1396    1408    1477    1523    1579    1664    1691    1760    1799    1876    1896
                1912    1918    1937    1977    1991    2006    2047    2072    2082    2109
.MACRO             1     385    1062
.MCALL           476
.NLIST             1     385     402     476     482     551     579     601     655     705     730     752     835     863     932
                 969    1040    1062    1063    1064    1065    1066    1067    1068    1069    1070    1071    1072    1073    1074
                1075    1076    1077    1078    1079    1080    1081    1082    1083    1084    1085    1086    1087    1088    1089
                1090    1091    1092    1093    1094    1095    1096    1097    1098    1099    1100    1101    1102    1103    1104
                1105    1106    1107    1108    1109    1110    1111    1112    1113    1114    1115    1116    1117    1118    1119
                1120    1121    1122    1123    1124    1125    1126    1127    1128    1129    1130    1131    1132    1133    1134
                1135    1136    1137    1138    1139    1140    1141    1142    1143    1144    1145    1146    1147    1148    1149
                1150    1151    1152    1153    1154    1155    1156    1157    1158    1159    1160    1161    1162    1163    1164
                1165    1166    1167    1168    1169    1170    1171    1172    1173    1174    1175    1176    1177    1178    1179
                1180    1181    1182    1183    1184    1185    1186    1187    1188    1189    1190    1191    1192    1193    1194
                1195    1196    1197    1198    1199    1200    1201    1202    1203    1204    1205    1206    1207    1208    1209
                1210    1211    1212    1213    1214    1215    1216    1217    1218    1219    1220    1221    1222    1223    1224
                1225    1226    1227    1228    1229    1230    1231    1232    1233    1234    1235    1236    1237    1238    1239
                1240    1241    1242    1243    1244    1245    1246    1247    1248    1249    1250    1251    1252    1253    1254
                1255    1256    1257    1258    1259    1260    1261    1262    1263    1264    1265    1266    1267    1268    1269
                1270    1271    1272    1273    1274    1275    1276    1277    1278    1279    1280    1281    1282    1283    1284
                1285    1286    1287    1288    1289    1290    1291    1292    1293    1294    1295    1296    1297    1298    1299
```

```
            1300    1301    1302    1303    1304    1305    1306    1307    1308    1309    1310    1311    1312    1313    1314
            1315    1316    1317    1330    1396    1408    1477    1523    1579    1664    1691    1760    1799    1876    1896
            1912    1918    1937    1977    1991    2006    2047    2072    2082    2109
.PAGE        478    1937    2006    2047    2082
.REM           1     402
.REPT        482    1062    1285
.SBTTL       385     402     551     579     601     655     705     738     752     835     863     932     969    1040    1330
            1396    1408    1477    1523    1579    1664    1691    1760    1799    1876    1896    1937    1977    2006    2047
            2082    2109
.TITLE       385
.WORD        717     731     869     871     873     875     877     879     881     883     885     887     889     891     893
             895     897     899     901     903     905     907     909     911     913     915     917     919     921     923
             925     927     929     931    1414    1416    1418    1420    1422    1424    1426    1428    1430    1432    1434
            1436    1438    1440    1442    1444    1446    1448    1450    1452    1454    1456    1458    1460    1462    1464
            1466    1468    1470    1472    1474    1476    1697    1699    1701    1703    1705    1707    1709    1711    1713
            1715    1717    1719    1721    1723    1725    1727    1729    1731    1733    1735    1737    1739    1741    1743
            1745    1747    1749    1751    1753    1755    1757    1759    2147
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZDJBD.SEQ/SOL/CRF=DZDJBD.MAC,DZDJBD.P11
RUN-TIME:  19 26 4 SECONDS
RUN-TIME RATIO: 238/50=4.6
CORE USED:  21K  (41 PAGES)
```

N05.

Spooler runtime 9 Seconds, 44 KCS, 234 disk reads, 3 disk writes, 64 pages

[redacted]