

# DJ11

EXERCISER  
MD-11-DZDJB-D

EP DZDJB-D-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 1

NOV 1976  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames. The leftmost column of frames contains text, likely labels for the data in the adjacent columns. The remaining frames contain numerical data organized into columns and rows, typical of a data table or ledger. The text in the first column is too small to read accurately but appears to include headers and possibly dates or identifiers.





MAINDEX-11-DJ11B-0-0  
TABLE OF CONTENTS

CONTENTS

1	ABSTRACT
1	REQUIREMENTS
1	EQUIPMENT
1	STORAGE
1	PRELIMINARY PROGRAMS
1	LOADING PROCEDURE
1	STARTING PROCEDURE
1	CONTROL SWITCH SETTINGS
1	STARTING ADDRESS
1	PROGRAM AND OPERATOR ACTION
1	OPERATING PROCEDURE
1	OPERATIONAL SWITCH SETTINGS
1	SUBROUTINE ABSTRACTS
1	PROGRAM AND OPERATOR ACTION
1	ERRORS
1	ERROR PRINTOUT
1	ERROR RECOVERY
1	ERROR COUNTER
1	RESTRICTIONS
1	MISCELLANEOUS
1	EXECUTION TIME
1	STACK POINTER
1	PASS COUNTER
1	POWER FAIL
1	PROGRAM DESCRIPTION

TABLE OF CONTENTS



MAINDEC-11-DZDJ8-D-C  
DZDJ8D.P11

DJ11 EXERCISER AND ON-LINE TESTS

**EO1**

MACY11 27(732) 21-SEP-76 13:54 PAGE 5

137  
138

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

139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194

MAINDEC-11-DZDJ8-D-D  
DESCRIPTION

DJ11 EXERCISER AND ON-LINE TESTS

PAGE 4

- 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 CNCE EVERY PASS.

5. OPERATING PROC

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
- SW<9> = 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. ↑ 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 (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY

MAINDEC-11-DZDJ8-D-0  
DZDJ8D.F11

DJ11 EXERCISER AND ON-LINE TESTS

GO1

MACY11 27(732) 21-SEP-76 13:54 PAGE 7

195  
:95

AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS  
END OF PASS, AND OTHER APPLICABLE AREAS.

197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000

MAINDEC-11-DZDJB-D-D  
DESCRIPTION

DJ11 EXERCISER AND ON-LINE TESTS

PAGE 5

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 (RD). 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 (RD) TO ALLOW THE OPERATOR TO SET THE DYNAMIC SWITCH SETTINGS (SEE SEC. 5.1).

6. ERRORS

6.1 ERROR PRINTOUT



MAINDEC-11-DZDJB-D-D  
DZDJB.D.P11

DJ11 EXERCISER AND ON-LINE TESTS

I01

MACY11 27(732) 21-SEP-76 13:54 PAGE 9

253  
254

THE FORMAT IS AS FOLLOWS:

255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310

MAINDEC-11-DZDJ8-D-D  
DESCRIPTION

DJ11 EXERCISER AND ON-LINE TESTS

PAGE 6

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 (DJ11'S WILL BE FIRST, STARTING AT 160010, THEN THE DZDJ8'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 VARIOUS BAUD RATES AVAILABLE AND THE ABILITY TO CHECK UP TO 8 DJ11'S AT ONCE, THE EXECUTION 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.

MAINDEC-11-DZDJB-D-D  
DZDJB0.P11

DJ11 EXERCISER AND ON-LINE TESTS

K01  
MACY11 27(732) 21-SEP-76 13:54 PAGE 11

311  
312

APPROX

313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368

MAINDEC-11-DZDJ8-D-D  
DESCRIPTION

DJ11 EXERCISER AND ON-LINE TESTS

PAGE 7

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 INITIALLY 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



MAINDEC-11-DZDJ8-D-D  
DZDJ8C.P11

DJ11 EXERCISER AND ON-LINE TESTS

MO1

MACY11 27(732) 21-SEP-76 13:54 PAGE 13

369  
370

RECEIVED ON THE RESPECTIVE LINE. A . . . . . (000) WILL "ECHO"  
72 TIMES AND THEN TURN OFF THE TRANSMITTER.

MAINDEC-11-DZDJB-D-D  
DESCRIPTION

DJ11 EXERCISER AND ON-LINE TESTS

PAGE 8

371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000

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

100000  
040000  
020000  
010000  
004000  
002000  
  
001000  
000400

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

DJ11 REGISTER BIT ASSIGNMENTS:

CONTROL STATUS REGISTER (CSR) XXXXXD  
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 INTERRUPT 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 INTERRUPT ENABLE (READ/WRITE)  
BIT15 TRANSMITTER READY (READ ONLY)

RECEIVER BUFFER REGISTER (RBUF) XXXXX2 (READ ONLY)

BIT0-7 RECEIVED CHARACTER









E02

MAINDEC-11-DZDJ8-D-D  
DZDJ8D.F11

DJ11 EXERCISER AND ON-LINE TESTS  
SET UP AREA

MACY11 27(732) 21-SEP-76 13:54 PAGE 18

```

551
552 005306 012706 001200 BEGIN: MOV #ICNT, SP ;SET UP STACK POINTER
553 005312 004737 014726 JSR PC,SUSWRR
554 005316 012700 003014 MOV #14, R0
555 005322 012720 014406 MOV #YESRT, (R0)+ ;TRACE TRAP VECTOR (14)
556 005326 012720 000340 MOV #340, (R0)+
557 005332 012720 014410 MOV #IOTRAP, (R0)+ ;IOT VECTOR (20)
558 005336 012720 000340 MOV #340, (R0)+
559 005342 012720 014246 MOV #PDOWNS, (R0)+ ;POWER FAIL VECTOR (24)
560 005346 012720 000340 MOV #340, (R0)+
561 005352 012720 013326 MOV #EMTS, (R0)+ ;EMT VECTOR (30)
562 005356 012720 000340 MOV #340, (R0)+
563 005362 012737 005420 0000 0 MOV #15, @#10
564 005370 006700 SXT R0
565 005372 012737 000006 014406 MOV #RTT, @#YESRT
566 005400 012737 000006 000006 MOV #RTT, @#6
567 005406 012737 003400 177774 MOV #400, @#177774 ;SET UP STACK LIMIT TO 1000
568 005414 005037 000006 CLR @#6
569 005420 012737 330012 000010 1$: MOV #12, @#10
570 005426 005037 001202 CLR ERRORS ;CLEAR ERROR COUNTER
571 005432 005037 001204 CLR PCNT ;CLEAR PASS COUNTER
572 005436 005037 001206 CLR PCNT+2
573 005442 012700 000300 MOV #300, R0 ;START OF FLOATING VECTOR AREA
574 005446 005720 2$: TST (R0)+ ;UPDATE POINTER
575 005450 010060 177776 MOV R0, -2(R0) ;PUT "+2" IN EACH VECTOR
576 005454 012720 000004 MOV #IOT, (R0)+ ;AND "IOT"
577 005460 022700 001000 CMP #1000, R0 ;CHECK FOR END OF FLOATING VECTOR AREA
578 005464 003370 BGT 2$ ;BRANCH IF MORE
579
580 :*****
581 :ROUTINE TO MAP ALL THE DJ11'S ON THE SYSTEM
582 :*****
583
584 005466 013700 001276 DJMAP: MOV FLTADR, R0 ;GET FIRST FLOATING ADDRESS
585 005472 012702 000001 MOV #1, R2 ;COUNTER FOR DJ11'S
586 005476 012737 000002 000006 MOV #RTI, @#6 ;RTI WHEN TIME-OUT
587 005504 005001 5$: CLR R1 ;SET UP COUNTER
588 005506 000261 1$: SEC ;SET CARRY
589 005510 005710 TST (R0) ;CHECK FOR A DEVICE
590 005512 103404 BCS 7$ ;BRANCH IF NONE
591 005514 062700 000010 6$: ADD #10, R0 ;GO TO NEXT DEVICE ADDRESS
592 005520 005201 INC R1 ;COUNT DJ11'S
593 005522 000771 BR 1$ ;LOOK FOR MORE
594
595 005524 005037 000006 7$: CLR @#6 ;RESTORE TIMEOUT VECTOR
596 005530 010137 001270 MOV R1, UNITS ;SAVE COUNT
597 005534 001005 BNE GETVEC
598 005536 000004 015254 TYPE, MSG01 ;TYPE "NO DJ11'S!"
599 005542 000000 HALT ;FATAL ERROR
600 005544 000137 013242 JMP @#DONE ;RESTART
601
602 :*****
603 :ROUTINE TO DETERMINE VECTOR ADDRESSES OF DJ11'S
604 :*****
605
606 005550 013746 000020 GETVEC: MOV @#20, -(SP) ;SAVE IOT VECTOR

```

```

607 005554 012737 005604 000020      MOV      #15, 2#20      ;RESET IOT VECTOR
608 005562 013701 001272      MOV      DEVADR, R1    ;FIRST DJ ADDRESS
609 005555 012711 040400      MOV      #40400, (R1) ;SET CSR
610                                     ;BIT8= TRANS SCAN ENABLE
611                                     ;BIT14= TRANS INTERRUPT ENABLE
612 005572 012761 000001 000004      MOV      #1, 4(R1)    ;TCR, LINE 0
613 005600 000001      WAIT                    ;WAIT FOR AN INTERRUPT
614 005602 000407      BR      3$             ;CONTINUE AFTER INTERUPT
615
616 005604 011502 1$:      MOV      (SP), R2      ;SAVE VECTOR ADR. (+4)
617 005606 152716 000010      SUB      #10, (SP)    ;REPOSITION ADR TO RCV. VEC.
618 005612 011637 001274      MOV      (SP), VECADR ;SAVE FIRST VECTOR
619 005616 022626      CMP      (SP)+, (SP)+ ;RESET STACK FROM IOT
620 005620 000002      RTI                    ;RETURN FROM INITIAL INTERUPT
621
622 005622 012637 000020 3$:      MOV      (SP)+, 2#20  ;RESTORE IOT VECTOR
623 005626 005742      TST     -(R2)          ;POINT TO XMT. VEC. +2
624 005630 013703 001270      MOV      UNITS, R3    ;SET UP UNIT COUNTER
625
626                                     ;CHECK THAT VECTORS ARE CONTIGUOUS
627
628 005634 005061 000004 2$:      CLR      4(R1)        ;CLEAR TCR
629 005640 005011      CLR      (R1)         ;CLEAR CSR
630 005642 012712 000004      MOV      #IOT, (R2)   ;RESTORE IOT TO XMT. VEC.+2
631 005646 005303      DEC      R3           ;CHECK FOR MORE DJ11'S
632 005650 001415      BEQ     REPORT        ;BRANCH IF DONE
633 005652 062701 000010      ADD      #10, R1      ;UPDATE DJ ADR. POINTER
634 005656 062702 000010      ADD      #10, R2      ;UPDATE VECTOR POINTER
635 005662 012712 000002      MOV      #RTI, (R2)   ;RTI ON INTERRUPT
636 005666 012711 040400      MOV      #40400, (R1) ;SET CSR
637 005672 012761 000001 000004      MOV      #1, 4(R1)   ;TCR LINE 0
638 005700 000001      WAIT                    ;WAIT FOR AN INTERRUPT
639 005702 000754      BR      2$
640
641                                     ;REPORT CONFIGURATION
642
643 005704 032777 020000 173276 REPORT: BIT #BIT13, 3SWR ;CHECK FOR INHIBIT TYPOLT
644 005712 001024      BNE     GETLEN        ;SKIP REPORT IF SET
645 005714 000004 015067      TYPE,   MSGMDN
646 005720 000004 015136      TYPE,   MSGADR
647 005724 013705 001272      MOV     DEVADR, TTY   ;TYPE DEVADR IN OCTAL
648 005730 004737 014064      JSR    PC, PRINTR    ;TYPE LEADING ZERO'S
649 005734 000004 015166      TYPE,   MSGVEC
650 005740 013705 001274      MOV     VECADR, TTY   ;TYPE VECADR IN OCTAL
651 005744 004737 014074      JSR    PC, PRINTS    ;AND SUPRESS LEADING ZERO'S
652 005750 000004 015212      TYPE,   MSGNUM
653 005754 013705 001270      MOV     UNITS, TTY    ;TYPE UNITS IN OCTAL
654 005760 004737 014074      JSR    PC, PRINTS    ;AND SUPRESS LEADING ZERO'S

```

```

655
656      :*****
657      :ROUTINE TO MAP CHARACTER LENGTHS
658      :*****
659
660 005764 022737 000176 001210 GETLEN: CMP      #SWREG, SWR
661 005772 001002          BNE      6$
662 005774 004737 014624          JSR      PC, CNLTLU
663 006000 013700 001270 6$:      MOV      UNITS, R0      ;SET UP UNIT COUNTER
664 006004 013701 001272          MOV      DEVADR, R1      ;SET UP DEVICE ADDRESS POINTER
665 006010 012702 000001 1$:      MOV      #1, R2      ;SET UP LINE MARKER
666 006014 012711 000415          MOV      #415, (R1)      ;RCV ENB, CMOS, MAINT., TRANS SCAN ENB
667 006020 032711 000020 10$:     BIT      #BIT4, (R1)      ;WAIT FOR MOS TO CLEAR
668 006024 001375          BNE      10$
669 006026 010261 000004 2$:      MOV      R2, 4(R1)      ;TRANS CONTROL, ONE LINE AT A TIME
670 006032 005711 3$:      TST      (R1)      ;WAIT FOR TRANS READY
671 006034 100376          BPL      3$
672 006036 012761 000377 000006  MOV      #377, 6(R1)      ;SEND A RUBOUT
673 006044 006302          ASL      R2      ;SKIP 4 LINES
674 006046 006302          ASL      R2
675 006050 006302          ASL      R2
676 006052 006302          ASL      R2
677 006054 103364          BCC      2$      ;BRANCH BACK IF MORE LINES
678 006056 005051 000004  CLR      4(R1)      ;CLEAR TCR
679 006062 062701 000010  ADD      #10, R1      ;UPDATE POINTER TO NEXT UNIT
680 006066 005300          DEC      #0
681 006070 001347          BNE      3$      ;CHECK FOR MORE UNITS
682 006072 013700 001270  MOV      UNITS, R0      ;SET UP UNIT COUNTER
683 006076 013701 001272          MOV      DEVADR, R1      ;SET UP DEVICE ADDRESS POINTER
684 006102 012702 004306          MOV      #MASK, R2      ;SET UP CHAR LEN TABLE POINTER
685 006106 012703 000004 4$:      MOV      #4, R3      ;SET UP CHAR COUNTER
686 006112 016104 000002 5$:      MOV      2(R1), R4      ;SAVE AND CHECK CHAR PRESENT
687 006116 100375          BPL      5$
688 006120 010405          MOV      R4, R5      ;DUP DATA
689 006122 000305          SWAB     R5      ;LINE # IN LOW BYTE
690 006124 042705 177760  BIC      #177760, R5      ;CLEAR ALL BUT LINE #
691 006130 006305          ASL      R5      ;*2
692 006132 060205          ADD      R2, R5      ;MAKE POINTER TO CHAR TABLE
693 006134 105104          COMB     R4      ;MAKE DATA INTO MASK
694 006136 042704 174000  BIC      #174000, R4      ;CLEAR UPPER BYTE
695 006142 010425          MOV      R4, (R5)+      ;SAVE THE MASK
696 006144 010425          MOV      R4, (R5)+      ;SAVE THE MASK
697 006146 010425          MOV      R4, (R5)+      ;SAVE THE MASK
698 006150 010425          MOV      R4, (R5)+      ;SAVE THE MASK
699 006152 005303          DEC      R3      ;COUNT TO 4
700 006154 001356          BNE      5$
701 006156 062701 000010  ADD      #10, R1      ;ADDRESS POINTER TO NEXT DJ
702 006162 062702 000040  ADD      #40, R2      ;CHAR LEN TABLE POINTER
703 006166 005300          DEC      R0      ;COUNT UNITS
704 006170 001346          BNE      4$      ;BRANCH BACK IF MORE

```



705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751

006172 005737 000042  
00617E 001051  
006200 000304 015235  
006204 004537 013520  
006210 006344  
006212 001367  
006214 013700 001270  
006220 012701 001222  
006224 012721 177777  
006230 005300  
006232 001374  
006234 032777 000400 172746  
006242 001427  
006244 005000  
006246 012701 001222  
006252  
006252 000004 006256  
006300 004537 013520  
006304 001214  
006306 013721 001214  
006312 005200  
006314 020037 001270  
006320 001354  
  
006322 013700 006344  
006326 001721  
006330 022700 000003  
006334 10371E  
006336 006300  
006340 000170 006344  
  
006344  
006344 000001  
006346 006354  
006350 010462  
006352 011666

```
*****
: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
*****

SELPRO: TST Q#42 ;CHECK FOR ACT 11 OR DDP
        BNE RESTAR ;BRANCH IF MONITOR
        TYPE, MSGPRG
        JSR RS, READIN ;READ A NUMBER FROM THE CTY
        .WORD PROGNO
        BNE SELPRO
        MOV UNITS, R0 ;SET UP UNIT COUNTER
        MOV #SVSW0,R1 ;SET UP SWITCH TABLE POINTER
1$: MOV #177777,(R1)+ ;SET ALL LINES
    DEC R0 ;COUNT UNITS
    BNE 1$ ;BRANCH IF MORE
    BIT #BITS, 2SWR ;CHECK FOR SW<8>, SELECT LINES
    BEQ RESTAR ;BRANCH IF NOT
    R0 ;SET UP UNIT COUNTER, DISPLAY
    MOV #SVSW0,R1 ;SET UP SWITCH TABLE POINTER

SWITCH: TYPE +2 ;.ASCIZ <15><12>"SELECT LINE = "
        JSR RS, READIN
        .WORD SAVIT
        MOV SAVIT,(R1)+
        INC R0 ;COUNT UNITS
        CMP R0, UNITS ;CHECK FOR MORE UNITS
        BNE SWITCH ;BRANCH IF MORE

.SBTTL RESTART POINT

RESTAR: MOV PROGNO, R0
        BEQ SELPRO
        CMP #3, R0
        BLO SELPRO
        ASL R0
        JMP 2PROGAD (R0)

PROGNO:
PROGAD: 1 ;DEFAULT TO PROGRAM 1
        PROG1
        PROG2
        PROG3
```



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	SVSWO(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	QSWR, 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	ADRNT(5), XMTTAB(4)	
823	076670	016564	007714 002306	MOV	ADRNT(5), RCVTAB(4)	
824	006615	116564	007716 003306	MOVB	CNTNIT(5), XMTCNT(4)	
825	006614	116564	007716 003307	MOVB	CNTNIT(5), RVCNT(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	001273	CMP	R0, UNITS	
833	006650	001273		BNE	11\$	
834	006652	042737	000140 177776	BIC	#140, Q#PS	;LOWER PROCESSOR PRIORITY

```

*****
:PROG1 BACKGROUND PROGRAM TO MONITOR TABLES
*****
: NOTE - PROGRAM MAY HANG IN A LOOP.
: IF THIS HAPPENS, RUN DZDJA.

```

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	Q#DONE	;SKIP ISR'S
863						

```

864
865
866
867
868 006756 004037 007472
869 006762 160012 000000
870 006766 004037 007356
871 006772 160010 000000
872 006776 004037 007472
873 007002 160022 000040
874 007006 004037 007356
875 007012 160020 000040
876 007016 004037 007472
877 007022 160032 000100
878 007026 004037 007356
879 007032 160030 000100
880 007036 004037 007472
881 007042 160042 000140
882 007046 004037 007356
883 007052 160040 000140
884 007056 004037 007472
885 007062 160052 000200
886 007066 004037 007356
887 007072 160050 000200
888 007076 004037 007472
889 007102 160062 000240
890 007106 004037 007356
891 007112 160060 000240
892 007116 004037 007472
893 007122 160072 000300
894 007126 004037 007356
895 007132 160070 000300
896 007136 004037 007472
897 007142 160102 000340
898 007146 004037 007356
899 007152 160100 000340
900 007156 004037 007472
901 007162 160112 000400
902 007166 004037 007356
903 007172 160110 000400
904 007176 004037 007472
905 007202 160122 000440
906 007206 004037 007356
907 007212 160120 000440
908 007216 004037 007472
909 007222 160132 000500
910 007226 004037 007356
911 007232 160130 000500
912 007236 004037 007472
913 007242 160142 000540
914 007246 004037 007356
915 007252 160140 000540
916 007256 004037 007472
917 007262 160152 000600
918 007266 004037 007356
919 007272 160150 000600

```

```

:*****
:PROG1 LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
:*****
RISR0: JSR      RO,RCVISR
        .WORD   <160012+<0*10>>,<40*0>
XISR0: JSR      RO,XMTISR
        .WORD   <160010+<0*10>>,<40*0>
RISR1: JSR      RO,RCVISR
        .WORD   <160012+<1*10>>,<40*1>
XISR1: JSR      RO,XMTISR
        .WORD   <160010+<1*10>>,<40*1>
RISR2: JSR      RO,RCVISR
        .WORD   <160012+<2*10>>,<40*2>
XISR2: JSR      RO,XMTISR
        .WORD   <160010+<2*10>>,<40*2>
RISR3: JSR      RO,RCVISR
        .WORD   <160012+<3*10>>,<40*3>
XISR3: JSR      RO,XMTISR
        .WORD   <160010+<3*10>>,<40*3>
RISR4: JSR      RO,RCVISR
        .WORD   <160012+<4*10>>,<40*4>
XISR4: JSR      RO,XMTISR
        .WORD   <160010+<4*10>>,<40*4>
RISR5: JSR      RO,RCVISR
        .WORD   <160012+<5*10>>,<40*5>
XISR5: JSR      RO,XMTISR
        .WORD   <160010+<5*10>>,<40*5>
RISR6: JSR      RO,RCVISR
        .WORD   <160012+<6*10>>,<40*6>
XISR6: JSR      RO,XMTISR
        .WORD   <160010+<6*10>>,<40*6>
RISR7: JSR      RO,RCVISR
        .WORD   <160012+<7*10>>,<40*7>
XISR7: JSR      RO,XMTISR
        .WORD   <160010+<7*10>>,<40*7>
RISR10: JSR     RO,RCVISR
        .WORD   <160012+<10*10>>,<40*10>
XISR10: JSR     RO,XMTISR
        .WORD   <160010+<10*10>>,<40*10>
RISR11: JSR     RO,RCVISR
        .WORD   <160012+<11*10>>,<40*11>
XISR11: JSR     RO,XMTISR
        .WORD   <160010+<11*10>>,<40*11>
RISR12: JSR     RO,RCVISR
        .WORD   <160012+<12*10>>,<40*12>
XISR12: JSR     RO,XMTISR
        .WORD   <160010+<12*10>>,<40*12>
RISR13: JSR     RO,RCVISR
        .WORD   <160012+<13*10>>,<40*13>
XISR13: JSR     RO,XMTISR
        .WORD   <160010+<13*10>>,<40*13>
RISR14: JSR     RO,RCVISR
        .WORD   <160012+<14*10>>,<40*14>
XISR14: JSR     RO,XMTISR
        .WORD   <160010+<14*10>>,<40*14>

```

```

920 007276 004037 007472
921 007302 160162 000640
922 007306 004037 007356
923 007312 160160 000640
924 007316 004037 007472
925 007322 160172 000700
926 007326 004037 007356
927 007332 160170 000700
928 007336 004037 007472
929 007342 160202 000740
930 007346 004037 007356
931 007352 160200 000740
932
933
934
935
936
937 007356
938 007356 010146
939 007360 010246
940 007362 012001
941 007364 007111
942 007366 100035
943 007370 116102 000007
944 007374 006302
945 007376 061002
946 007400 105762 003306
947 007404 001410
948 007406 117261 001306 000006
949 007414 105362 003306
950 007420 005262 001306
951 007424 000757
952 007426 010346
953 007430 005062 001306
954 007434 161002
955 007436 006202
956 007440 005003
957 007442 000261
958 007444 006103
959 007446 005302
960 007450 100375
961 007452 040361 000004
962 007456 012603
963 007460 000741
964 007462
965 007462 012602
966 007464 012601
967 007466 012600
968 007470 000002
969
970
971
972
973
974 007472
975 007472 010146

```

```

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

```

```

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

```

```

XMTISR:
        MOV R1,-(6) ;PUSH R1 ON STACK
        MOV R2,-(6) ;PUSH R2 ON STACK
        MOV (R0)+,R1
1$:     TST (R1) ;CHECK FOR TRANS READY
        BPL 4$
        MOVB 7(R1),R2 ;GET LINE NO.
        ASL R2
        ADD (R0),R2
        TSTB XMTCNT(2) ;TST FOR ZERO
        BEQ 2$
        MOVB @XMTTAB(2),6(1) ;SEND A CHARACTER
        DECB XMTCNT(2) ;COUNT CHARACTERS
        INC XMTTAB(2) ;UPDATE TABLE POINTER
        BR 1$
2$:     MOV R3,-(SP)
        CLRB XMTTAB(2) ;CLEAR TABLE POINTER
        SUB (R0),R2
        ASR R2
        CLRB R3
3$:     ROL R3
        DEC R2
        BPL 3$
        BIC R3,4(R1) ;CLEAR TCR BIT FOR LINE
        MOV (SP)+,R3 ;RESTORE R3
        BR 1$
4$:     MOV (6)+,R2 ;POP STACK INTO R2
        MOV (6)+,R1 ;POP STACK INTO R1
        MOV (6)+,R0 ;POP STACK INTO R0
        RTI

```

```

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

```

```

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

```

```

976 007474 010246      MOV      R2,-(6)      ;PUSH R2 ON STACK
977 007476 010346      MOV      R3,-(6)      ;PUSH R3 ON STACK
978 007500 010446      MOV      R4,-(6)      ;PUSH R4 ON STACK
979 007502 012001      MOV      (R0)+,R1     ;GET RBUF ADDRESS
980 007504 011102      1$:     MOV      (R1),R2     ;READ THE DATA
981 007506 100074      BPL      7$           ;BRANCH IF NO CHAR PRESENT
982 007510 032702 070000      BIT      #70000,R2   ;CHECK FOR ERRORS
983 007514 001403      BEQ      2$           ;BRANCH IF OK
984 007516 104002      HLT+2              ;RECEIVER ERROR
985                          ;R1=RBUF ADDRESS
986                          ;R2=CONTENTS OF RBUF
987                          ;BIT12=PARITY ERROR
988                          ;BIT13=FRAMING ERROR
989                          ;BIT14=UART OVERRUN
990 007520 042702 070000      BIC      #70000, R2   ;CLEAR ERROR BITS FOR SPEED
991 007524 010204      2$:     MOV      R2, R4      ;DUP THE RBUF
992 007526 105004      CLRB    R4           ;CLEAR THE DATA
993 007530 000304      SWAB    R4           ;LINE # TO LOW BYTE
994 007532 106304      ASLB    R4           ;LINE # * 2, ALSO CLR CHAR PRESENT
995 007534 061004      ADD     (R0),R4      ;ADD OFFSET
996 007536 117403 002306      MOVB    @RCVTAB(R4),R3 ;GET EXPECTED DATA
997 007542 046403 004306      BIC     MASK(4),R3   ;MASK CHARACTER LENGTH
998 007546 120302      CMPB    R3,R2
999 007550 001403      BEQ     3$           ;BRANCH IF OK
1000 007552 042703 177400      BIC     #177400,R3   ;MAKE SURE UPPER BYTE CLEAR
1001 007556 104003      HLT+3              ;DATA ERROR
1002                          ;R1=RBUF ADDRESS
1003                          ;R2=CONTENTS OF RBUF(DATA)
1004                          ;R3=EXPECTED DATA
1005 007560 105364 003307      3$:     DECB    RCVCNT(4)
1006 007564 001403      BEQ     4$
1007 007566 005264 002306      INC     RCVTAB(4)   ;UPDATE TABLE POINTER
1008 007572 000744      BR      1$           ;CONTINUE
1009 007574 105764 004307      4$:     TSTB    CNTTAB(4) ;CHECK FOR DONE 8 MESSAGES
1010 007600 100741      BMI     1$           ;BRANCH IF DONE (DON'T RESTART TRANSMITTER)
1011 007602 001740      BEQ     1$           ;BRANCH IF DONE (DON'T RESTART TRANSMITTER)
1012                          ;NOTE: IF THE ABOVE LOCATION IS PATCHED TO 1402, THE FAST LINES WILL
1013                          ;CONTINUE TO RUN UNTIL ALL LINES HAVE COMPLETED 8 MESSAGES.
1014                          ;THE WAY IT IS CODED NOW, ALL LINES COMPLETE JUST 8 MESSAGES.
1015 007604 105364 004307      5$:     DECB    CNTTAB(4)
1016 007610 017703 171374      MOV     @SWR,R3     ;GET SWITCH SETTINGS
1017 007614 042703 177770      BIC     #177770,R3   ;MASK SW<2:0>
1018 007620 006303      ASL     R3
1019 007622 006303      ASL     R3
1020 007624 016364 007714 001306      MOV     @ADRNT(3),XMTTAB(4)
1021 007632 016364 007714 002306      MOV     @ADRNT(3),RCVTAB(4)
1022 007640 116364 007716 003306      MOVB    CNTNIT(3),XMTCNT(4)
1023 007646 116364 007716 003307      MOVB    CNTNIT(3),RCVCNT(4)
1024 007654 161004      SUB     (R0),R4
1025 007656 006204      ASR     R4
1026 007660 005003      CLR     R3
1027 007662 000261      SEC
1028 007664 006103      6$:     ROL     R3
1029 007666 005304      DEC     R4
1030 007670 100375      BPL     6$
1031 007672 050361 000002      BIS     R3,2(R1)    ;SET TCR BIT FOR LINE

```

```

1032 007576 000702
1033 007700
1034 007790 012604
1035 007702 012603
1036 007704 012602
1037 007706 012601
1038 007710 012600
1039 007712 000002
1040
1041
1042
1043
1044
1045 007714 007760
1046 007716 000377
1047 007720 010360
1048 007722 000100
1049 007724 007752
1050 007726 000106
1051 007730 015360
1052 007732 000001
1053 007734 015760
1054 007736 000001
1055 007740 016360
1056 007742 000001
1057 007744 016760
1058 007746 000001
1059 007750 017360
1060 007752 005015 177777 177777
1061 007760 040
1062 007761 041
1063 007762 042
1064 007763 043
1065 007764 044
1066 007765 045
1067 007766 046
1068 007767 047
1069 007770 050
1070 007771 051
1071 007772 052
1072 007773 053
1073 007774 054
1074 007775 055
1075 007776 056
1076 007777 057
1077 010000 060
1078 010001 061
1079 010002 062
1080 010003 063
1081 010004 064
1082 010005 065
1083 010006 066
1084 010007 067
1085 010010 070
1086 010011 071
1087 010012 072

```

```

7S: BR IS
MOV (6)+,R4 ;POP STACK INTO R4
MOV (6)+,R3 ;POP STACK INTO R3
MOV (6)+,R2 ;POP STACK INTO R2
MOV (6)+,R1 ;POP STACK INTO R1
MOV (6)+,R0 ;POP STACK INTO R0
RTI

```

```

*****
:PROG1 DATA TABLES
*****

```

```

ADMNIT: BINARY ;SW<2:0>=0 BINARY COUNT PATTERN
CNTNIT: 377 ;SIZE=256.
PHRASE ;SW<2:0>=1 "THE QUICK BROW" FOX..."
64 ;SIZE=64.
SIXBIT ;SW<2:0>=2 040 THRU 137
70 ;SIZE=70.
END

```

```

|
| END+400
|
| END+1000
|
| END+1400
|
| END+2000
SIXBIT: .ASCII (15)<(12)<(377)<(377)<(377)<(377) ;CR-LF, FILLERS
BINARY: .BYTE 40
        .BYTE 41
        .BYTE 42
        .BYTE 43
        .BYTE 44
        .BYTE 45
        .BYTE 46
        .BYTE 47
        .BYTE 50
        .BYTE 51
        .BYTE 52
        .BYTE 53
        .BYTE 54
        .BYTE 55
        .BYTE 56
        .BYTE 57
        .BYTE 60
        .BYTE 61
        .BYTE 62
        .BYTE 63
        .BYTE 64
        .BYTE 65
        .BYTE 66
        .BYTE 67
        .BYTE 70
        .BYTE 71
        .BYTE 72

```







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	05.517	020116	047506			
1322	0104.0	020130	052512	050115			
1323	010416	042105	047440	042526			
1324	010424	020122	026077	033470			
1325	010432	026066	032065	026063			
1326	010440	030462	027050	020060			
1327	010446	040514	054532	042040			
1328	010454	043517	020523	000			
1329		010462				.EVEN	

G03

MAINDEC-11-DZDJB-D-D  
DZDJB0.P11

DJ11 EXERCISER AND ON-LINE TESTS  
ON-LINE EXERCISER (TRANSMIT LAST CHARACTER RECEIVED)

MACY11 27(732) 21-SEP-76 13:54 PAGE 33

1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1339 010462 000005  
1339 010464 012706 001200  
1340 010470 052737 000340 177776  
1341 010476 012701 001306  
1342 010502 012702 001400  
1343 010506 005021  
1344 010510 005302  
1345 010512 001375  
1346 010514 012702 000400  
1347 010520 005201  
1348 010522 105021  
1349 010524 005302  
1350 010526 001374  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358 010530 005000  
1359 010532 013701 001272  
1360 010536 013702 001274  
1361 010542 012703 010720  
1362 010546 010322  
1363 010550 013722 001302  
1364 010554 022323  
1365 010556 010113  
1366 010560 062723 000002  
1367 010564 005723  
1368 010566 010322  
1369 010570 013722 001304  
1370 010574 022323  
1371 010576 010123  
1372 010600 012721 050501  
1373  
1374  
1375  
1376  
1377  
1378 010604 005721  
1379 010606 006300  
1380 010610 016011 001222  
1381 010614 006200  
1382 010616 012737 000001 001262  
1383 010624 012304  
1384 010626 033711 001262  
1385 010632 001406

```

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

PROG2:  RESET                ;CLEAR OUT THE WORLD
        MOV      #ICNT, SP   ;RESET THE STACK POINTER
        BIS      #340, 0#PS ;PROCESSOR TO LEVEL 7
        MOV      #XMTTAB,R1 ;FIRST TABLE POINTER
        MOV      #1400, R2  ;LENGTH OF TABLES (WORDS)
1$:     CLR      (R1)+       ;CLEAR THE TABLE
        DEC      R2
        BNE     1$
        MOV      #400,R2    ;LENGTH OF MASK/COUNT TABLE
2$:     INC      R1          ;SKIP MASK
        CLRB    (R1)+       ;CLEAR COUNT
        DEC      R2
        BNE     2$

;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

P2INIT: CLR      R0
        MOV      DEVADR, R1
        MOV      VECADR, R2
        MOV      #R2SRO,R3
1$:     MOV      R3, (R2)+   ;SET UP POINTER TO LINKERS
        MOV      RCVLVL, (R2)+ ;SET UP RECEIVER INTERRUPT VECTOR
        CMP      (R3)+, (R3)+ ;ADD 4 TO R3
        MOV      R1, (R3)    ;ADDRESS OF CSR
        ADD      #2, (R3)+   ;ADDRESS OF RBUF
        TST      (R3)+
        MOV      R3, (R2)+   ;SET UP TRANSMITTER INTERRUPT VECTOR
        MOV      XMTLVL, (R2)+
        CMP      (R3)+, (R3)+
        MOV      R1, (R3)+   ;ADDRESS OF CSR
        MOV      #50501, (R1)+ ;SET UP CSR
        ;BIT0 = RECEIVER ENABLE
        ;BIT6 = RECEIVER INTERRUPT ENABLE
        ;BIT8 = TRANSMITTER SCAN ENABLE
        ;BIT12 = STATUS ENABLE
        ;BIT14 = TRANSMITTER INTERRUPT ENABLE

        TST      (R1)+
        ASL      R0          ;UNIT * * 2
        MOV      SVSWO(R0), (R1) ;SET TCR BITS FOR SELECTED LINES
        ASR      R0          ;RESET UNIT COUNTER
        MOV      #1, MARK    ;SET UP MARKER
        MOV      (R3)+, R4   ;SET UP OFFSET TO TABLES
4$:     BIT      MARK, (R1)  ;CHECK FOR LINE SELECTED
        BEQ     5$

```

# H03

MAINDEC-11-DZDJ8-D-D  
DZDJ8D.P11

PROG2:

DJ11 EXERCISER AND ON-LINE TESTS  
ON-LINE EXERCISER (TRANSMIT LAST CHARACTER RECEIVED)

MACY11 27(732) 21-SEP-76 13:54 PAGE 34

```
1386 010634 012764 015271 001306      MOV      #MSGP2, XMRTAB(4) ;SET UP XMTR TABLE
1387 010642 012764 000045 003306      MOV      #45, XMTCNT(4) ;SET UP COUNT
1388 010650 005724          SS:      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      RO
1393 010664 020037 001270      CMP      RO, UNITS
1394 010670 001326          BNE      1$
1395 010672 042737 000140 177776      BIC      #140, @#PS ;LOWER PROCESSOR PRIORITY
1396
1397
1398          :*****
1399          :PROG2 FOREGROUND PROGRAM TO READ/WRITE MEMORY
1400          :*****
1401 010700 012700 020000      FORP2:  MOV      #20000,RO ;TOP OF 4K BANK OF MEMORY
1402 010704 000241          CLC
1403 010706 005540          1$:    ADC      -(RO) ;FAST READ/WRITE TO MEMORY
1404 010710 001376          BNE      1$ ;RAPID REPEAT
1405 010712 005700          TST      RO ;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      RO,P2RISR
1414 010724 160012 000000          .WORD   <160012+(0*10)>,<0*40>
1415 010730 004037 011320      X2SR0: JSR      RO,P2XISR
1416 010734 160020 000000          .WORD   <160020+(0*10)>,<0*40>
1417 010740 004037 011512      R2SR1: JSR      RO,P2RISR
1418 010744 160022 000040          .WORD   <160012+(1*10)>,<1*40>
1419 010750 004037 011320      X2SR1: JSR      RO,P2XISR
1420 010754 160020 000040          .WORD   <160020+(1*10)>,<1*40>
1421 010760 004037 011512      R2SR2: JSR      RO,P2RISR
1422 010764 160032 000100          .WORD   <160012+(2*10)>,<2*40>
1423 010770 004037 011320      X2SR2: JSR      RO,P2XISR
1424 010774 160040 000100          .WORD   <160020+(2*10)>,<2*40>
1425 011000 004037 011512      R2SR3: JSR      RO,P2RISR
1426 011004 160042 000140          .WORD   <160012+(3*10)>,<3*40>
1427 011010 004037 011320      X2SR3: JSR      RO,P2XISR
1428 011014 160050 000140          .WORD   <160020+(3*10)>,<3*40>
1429 011020 004037 011512      R2SR4: JSR      RO,P2RISR
1430 011024 160052 000200          .WORD   <160012+(4*10)>,<4*40>
1431 011030 004037 011320      X2SR4: JSR      RO,P2XISR
1432 011034 160060 000200          .WORD   <160020+(4*10)>,<4*40>
1433 011040 004037 011512      R2SR5: JSR      RO,P2RISR
1434 011044 160062 000240          .WORD   <160012+(5*10)>,<5*40>
1435 011050 004037 011320      X2SR5: JSR      RO,P2XISR
1436 011054 160070 000240          .WORD   <160020+(5*10)>,<5*40>
1437 011060 004037 011512      R2SR6: JSR      RO,P2RISR
1438 011064 160072 000300          .WORD   <160012+(6*10)>,<6*40>
1439 011070 004037 011320      X2SR6: JSR      RO,P2XISR
1440 011074 160100 000300          .WORD   <160020+(6*10)>,<6*40>
1441 011100 004037 011512      R2SR7: JSR      RO,P2RISR
```



```

1442 011104 160102 000340
1443 011110 004037 011320
1444 011114 160110 000340
1445 011120 004037 011512
1446 011124 160112 000400
1447 011130 004037 011320
1448 011134 160120 000400
1449 011140 004037 011512
1450 011144 160122 000440
1451 011150 004037 011320
1452 011154 160130 000440
1453 011160 004037 011512
1454 011164 160132 000500
1455 011170 004037 011320
1456 011174 160140 000500
1457 011200 004037 011512
1458 011204 160142 000540
1459 011210 004037 011320
1460 011214 160150 000540
1461 011220 004037 011512
1462 011224 160152 000600
1463 011230 004037 011320
1464 011234 160160 000600
1465 011240 004037 011512
1466 011244 160162 000640
1467 011250 004037 011320
1468 011254 160170 000640
1469 011260 004037 011512
1470 011264 160172 000700
1471 011270 004037 011320
1472 011274 160200 000700
1473 011300 004037 011512
1474 011304 160202 000740
1475 011310 004037 011320
1476 011314 160210 000740
1477
1478
1479
1490
1481
1482 011320
1483 011320 010146
1484 011322 010246
1485 011324 012001
1486 011326 005711
1487 011330 100064
1488 011332 116102 000007
1489 011336 006302
1490 011340 061002
1491 011342 105762 003306
1492 011346 001413
1493 011350 117261 001306 000006
1494 011356 105362 003306
1495 011362 105762 004307
1496 011366 001357
1497 011370 005262 001306

```

```

        .WORD <160012+(7*10)>,<7*40>
X2SR7: JSR   RO,P2XISR
        .WORD <160020+(7*10)>,<7*40>
R2SR10: JSR  RO,P2RISR
        .WORD <160012+(10*10)>,<10*40>
X2SR10: JSR  RO,P2XISR
        .WORD <160020+(10*10)>,<10*40>
R2SR11: JSR  RO,P2RISR
        .WORD <160012+(11*10)>,<11*40>
X2SR11: JSR  RO,P2XISR
        .WORD <160020+(11*10)>,<11*40>
R2SR12: JSR  RO,P2RISR
        .WORD <160012+(12*10)>,<12*40>
X2SR12: JSR  RO,P2XISR
        .WORD <160020+(12*10)>,<12*40>
R2SR13: JSR  RO,P2RISR
        .WORD <160012+(13*10)>,<13*40>
X2SR13: JSR  RO,P2XISR
        .WORD <160020+(13*10)>,<13*40>
R2SR14: JSR  RO,P2RISR
        .WORD <160012+(14*10)>,<14*40>
X2SR14: JSR  RO,P2XISR
        .WORD <160020+(14*10)>,<14*40>
R2SR15: JSR  RO,P2RISR
        .WORD <160012+(15*10)>,<15*40>
X2SR15: JSR  RO,P2XISR
        .WORD <160020+(15*10)>,<15*40>
R2SR16: JSR  RO,P2RISR
        .WORD <160012+(16*10)>,<16*40>
X2SR16: JSR  RO,P2XISR
        .WORD <160020+(16*10)>,<16*40>
R2SR17: JSR  RO,P2RISR
        .WORD <160012+(17*10)>,<17*40>
X2SR17: JSR  RO,P2XISR
        .WORD <160020+(17*10)>,<17*40>

```

```

;*****
;PROG2 TRANSMITTER INTERRUPT SERVICE ROUTINE
;*****

```

```

P2XISR:
        MOV   R1,-(6)           ;PUSH R1 ON STACK
        MOV   R2,-(6)           ;PUSH R2 ON STACK
        MOV   (R0)+,R1
is:     TST   (R1)               ;CHECK FOR TRANS READY
        BPL   4$
        MOVB  7(R1),R2           ;GET LINE NO.
        ASL   R2
        ADD   (R0),R2
        TSTB  XMTCNT(2)         ;TST FOR ZERO
        BEQ   2$                 ;GET OUT
        MOVB  @XMTTAB(2),5(R1)  ;SEND A CHARACTER
        DECB  XMTCNT(2)         ;COUNT CHARACTERS
        TSTB  CNTTAB(2)        ;CHECK FOR MESSAGE OR DATA
        BNE  1$                 ;BRANCH IF DATA
        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 001306  ADD    R2, XMTTAB(2)    ;F 3 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          SJB    (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$:
1519 011502 012602          MOV    (6)+,R2      ;POP STACK INTO R2
1520 011504 012601          MOV    (6)+,R1      ;POP STACK INTO R1
1521 011506 012600          MOV    (6)+,R0      ;POP STACK INTO R0
1522 011510 000002          RTI
1523
1524          ;*****
1525          ;PROG2 RECEIVER INTERRUPT SERVICE ROUTINE
1526          ;*****
1527
1528 011512          P2RISR:
1529 011512 010146          MOV    R1,-(6)      ;PUSH R1 ON STACK
1530 011514 010246          MOV    R2,-(6)      ;PUSH R2 ON STACK
1531 011516 010346          MOV    R3,-(6)      ;PUSH R3 ON STACK
1532 011520 012001          MOV    (R0)+,R1     ;GET RBUF ADDRESS
1533 011522 011102 15:  MOV    (R1),R2      ;READ THE DATA
1534 011524 100053          BPL    7$          ;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+c          ;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$:  MOV    R2, R3      ;DUP THE RBUF
1546 011542 105003          CLRB  R3          ;CLEAR THE DATA
1547 011544 000303          SWAB  R3          ;LINE # TO LOW BYTE
1548 011546 106303          ASLB  R3          ;LINE # * 2. ALSO CLR CHAR PRESENT
1549 011550 061003          ADD    (R0),R3     ;ADD OFFSET
1550 011552 136302 004306  BITB  MASK(3),R2    ;CHECK CHARACTER LENGTH
1551 011556 00.01          BEQ    3$          ;BRANCH IF OK
1552 011560 104002          HLT+2          ;CHARACTER LENGTH ERROR
1553          ;R1=RBUF ADDRESS

```

# K03

MAINDEC-11-DZDJ8-D-D  
DZDJ8D.P11

DJ11 EXERCISER AND ON-LINE TESTS  
RECEIVER ISR

MACY11 27(732) 21-SEP-76 13:54 PAGE 37

```

1554                                     ;R2=CONTENTS OF RBUF(DATA)
1555 011562 105763 002306      3$:  TSTB  RCVTAB(3)      ;CHECK FOR BREAK
1556 011566 001017                                     ;BRANCH IF REAL DATA
1557 011570 110263 002306      MOVB  R2,    RCVTAB(3) ;SAVE THE DATA
1558 011574 151003                                     ;RECOVER LINE NUMBER
1559 011576 006203      ASR    R3
1560 011600 005037 001262      CLR    MARK          ;SET UP MARKER
1561 011604 000261      SEC
1562 011606 006137 001262      4$:  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 011654 000002      RTI

```



# M03

MAINDEC-11-DZDJ8-D-0  
DZDJ8D.P11

DJ11 EXERCISER AND ON-LINE TESTS  
EC40 EXERCISER

MACY11 27(732) 21-SEP-76 13:54 PAGE 39

```

1635 012064 006200          ASR      R0          ;RESET UNIT COUNTER
1636 012066 012737 000001 001262  MOV     #1          ;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          MOVVB  (R3)+, (R5)+ ;MOVE MESSAGE INTO BUFFER
1647 012132 001404          SEQ     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 012148 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$:  BIS     #101, (R1) ;SET RECEIVER ENABLES OF CSR
1679
1680 ;BIT0 = RECEIVER ENABLE
1681 ;BIT6 = RCV INTERRUPT ENABLE
1681 012244 062701 000010  ADD     #10, R1    ;UPDATE TO NEXT DJ11
1682 012250 005300          DEC     R0         ;COUNT DJ11'S
1683 012252 001372          BNE     ?$         ;
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
1694
1695
1696 012274 004037 013024
1697 012300 160012 000000
1698 012304 004037 012674
1699 012310 160020 000000
1700 012314 004037 013024
1701 012320 160022 000040
1702 012324 004037 012674
1703 012330 160030 000040
1704 012334 004037 013024
1705 012340 160032 000100
1706 012344 004037 012674
1707 012350 160040 000100
1708 012354 004037 013024
1709 012360 160042 000140
1710 012364 004037 012674
1711 012370 160050 000140
1712 012374 004037 013024
1713 012400 160052 000200
1714 012404 004037 012674
1715 012410 160060 000200
1716 012414 004037 013024
1717 012420 160062 000240
1718 012424 004037 012674
1719 012430 160070 000240
1720 012434 004037 013024
1721 012440 160072 000300
1722 012444 004037 012674
1723 012450 160100 000300
1724 012454 004037 013024
1725 012460 160102 000340
1726 012464 004037 012674
1727 012470 160110 000340
1728 012474 004037 013024
1729 012500 160112 000400
1730 012504 004037 012674
1731 012510 160120 000400
1732 012514 004037 013024
1733 012520 160122 000440
1734 012524 004037 012674
1735 012530 160130 000440
1736 012534 004037 013024
1737 012540 160132 000500
1738 012544 004037 012674
1739 012550 160140 000500
1740 012554 004037 013024
1741 012560 160142 000540
1742 012564 004037 012674
1743 012570 160150 000540
1744 012574 004037 013024
1745 012600 160152 000600
1746 012604 004037 012674

```

```

;*****
:PROG3 LINKERS TO DJ11 INTERRUPT SERVICE ROUTINES
;*****

```

```

R3SR0: JSR RO,P3RISR
        .WORD <160012+<0*10>>,<0*40>
X3SR0: JSR RO,P3XISR
        .WORD <160020+<0*10>>,<0*40>
R3SR1: JSR RO,P3RISR
        .WORD <160012+<1*10>>,<1*40>
X3SR1: JSR RO,P3XISR
        .WORD <160020+<1*10>>,<1*40>
R3SR2: JSR RO,P3RISR
        .WORD <160012+<2*10>>,<2*40>
X3SR2: JSR RO,P3XISR
        .WORD <160020+<2*10>>,<2*40>
R3SR3: JSR RO,P3RISR
        .WORD <160012+<3*10>>,<3*40>
X3SR3: JSR RO,P3XISR
        .WORD <160020+<3*10>>,<3*40>
R3SR4: JSR RO,P3RISR
        .WORD <160012+<4*10>>,<4*40>
X3SR4: JSR RO,P3XISR
        .WORD <160020+<4*10>>,<4*40>
R3SR5: JSR RO,P3RISR
        .WORD <160012+<5*10>>,<5*40>
X3SR5: JSR RO,P3XISR
        .WORD <160020+<5*10>>,<5*40>
R3SR6: JSR RO,P3RISR
        .WORD <160012+<6*10>>,<6*40>
X3SR6: JSR RO,P3XISR
        .WORD <160020+<6*10>>,<6*40>
R3SR7: JSR RO,P3RISR
        .WORD <160012+<7*10>>,<7*40>
X3SR7: JSR RO,P3XISR
        .WORD <160020+<7*10>>,<7*40>
R3SR10: JSR RO,P3RISR
        .WORD <160012+<10*10>>,<10*40>
X3SR10: JSR RO,P3XISR
        .WORD <160020+<10*10>>,<10*40>
R3SR11: JSR RO,P3RISR
        .WORD <160012+<11*10>>,<11*40>
X3SR11: JSR RO,P3XISR
        .WORD <160020+<11*10>>,<11*40>
R3SR12: JSR RO,P3RISR
        .WORD <160012+<12*10>>,<12*40>
X3SR12: JSR RO,P3XISR
        .WORD <160020+<12*10>>,<12*40>
R3SR13: JSR RO,P3RISR
        .WORD <160012+<13*10>>,<13*40>
X3SR13: JSR RO,P3XISR
        .WORD <160020+<13*10>>,<13*40>
R3SR14: JSR RO,P3RISR
        .WORD <160012+<14*10>>,<14*40>
X3SR14: JSR RO,P3XISR

```





```

1903
1904 013024 P3RISR:
1905 013024 010146 MOV R1,-(6) ;PUSH R1 ON STACK
1906 013026 010246 MOV R2,-(6) ;PUSH R2 ON STACK
1907 013030 010346 MOV R3,-(6) ;PUSH R3 ON STACK
1908 013032 010446 MOV R4,-(6) ;PUSH R4 ON STACK
1909 013034 012001 MOV (R0)+,R1 ;GET RBUF ADDRESS
1910 013036 011102 1S: MOV (R1),R2 ;READ THE DATA
1911 013040 100072 BPL 8S ;BRANCH IF NO CHAR PRESENT
1912 013042 032702 070000 BIT #70000,R2 ;CHECK FOR ERRORS
1913 013046 001402 BEQ 2S ;BRANCH IF OK
1914 013050 104002 HLT+2 ;RECEIVER ERROR
1915 ;R1=RBUF ADDRESS
1916 ;R2=CONTENTS OF RBUF
1917 ;BIT12=PARITY ERROR
1918 ;BIT13=FRAMING ERROR
1919 ;BIT14=UART OVERRUN
1920 ;SKIP BAD DATA
1921 013052 000771 2S: BR 1S ;DUP THE RBUF
1922 013054 070204 MOV R2,R4 ;CLEAR THE DATA
1923 013056 105004 CLRB R4 ;LINE # TO LOW BYTE
1924 013060 000304 SWAB R4 ;LINE # * 2, ALSO CLR CHAR PRESENT
1925 013062 10E304 ASLB R4 ;ADD OFFSET
1926 013064 061004 ADD (R0),R4 ;CHECK CHARACTER LENGTH
1927 013066 136402 004306 BITB MASK(4),R2 ;BRANCH IF OK
1928 013072 001401 BEQ 3S ;CHARACTER LENGTH ERROR
1929 013074 104002 HLT+2 ;R1=RBUF ADDRESS
1930 ;R2=CONTENTS OF RBUF(DATA)
1931 013076 005764 3S: TST RCVTAB(4) ;CHECK FOR UNSELECTED LINE
1932 013102 001002 BNE 4S ;BRANCH IF OK
1933 013104 104002 HLT+2 ;RECEIVED DATA ON UNSELECTED LINE
1934 ;R1 = RBUF ADDRESS
1935 ;R2 = CONTENTS OF RBUF
1936 013106 000753 4S: BR 1S ;IGNORE THE DATA
1937 013110 105774 002306 TSTB JRCVTAB(4) ;CHECK FOR DATA BUFFER FULL
1938 013114 001403 BEQ 5S ;BRANCH IF OK
1939 013116 104002 HLT+2 ;SOFTWARE DATA BUFFER OVERFLOW
1940 ;POSSIBLE TRANSMITTER PROBLEM
1941 ;R1 = RBUF ADDRESS
1942 ;R2 = CONTENTS OF RBUF
1943 ;NOTE: IF THE ABOVE ERROR WAS DUE TO OVERLOAD, INCREASING THE CONTENTS
1944 ; OF "BUFSIZ" MAY RECTIFY THE PROBLEM.
1945 ; "BUFSIZ" MUST BE A MULTIPLE OF 2.
1946 ; INCREASING IT MAY CAUSE THE BUFFERS TO OVERFLOW 4K.
1947 013120 000137 011666 JMP PROG3 ;RESTART ON THIS TYPE ERROR
1948 013124 005764 5S: TST XMTTAB(4) ;CHECK FOR TRANSMITTER ACTIVE
1949 013130 001414 BEQ 6S ;BRANCH IF INACTIVE
1950 013132 110274 002306 MOV R2,JRCVTAB(4) ;PUT THE DATA IN THE BUFFER
1951 013136 005264 002306 INC RCVTAB(4) ;UPDATE POINTER TO NEXT SPACE
1952 013142 033764 001266 002306 BIT BUFMSK,RCVTAB(4) ;CHECK FOR END OF BUFFER
1953 013150 001332 BNE 1S ;BRANCH IF NOT
1954 013152 163764 001264 002306 SUB BUFSIZ,RCVTAB(4) ;RAP AROUND BUFFER
1955 013160 000726 BR 1S
1956 013162 042764 001266 002306 6S: BIC BUFMSK,RCVTAB(4) ;RESET TABLE POINTER
1957 013170 016464 002306 001306 MOV RCVTAB(4),XMTTAB(4)

```

```

:359 013176 110274 002306      MOVB   R2, JRCVTAB(4)
:360 013202 161004      SUB    (R0), R4
:361 013204 006204      ASRA  R4
:362 013206 005003      CLRA  R3
:363 013210 000261      SEC
:364 013212 006103      7$:   ROL   R3
:365 013214 005304      DEC   R4
:366 013216 100375      BPL   7$
:367 013220 050361 000002      BIS   R3, 2(R1)      :SET TCR BIT FOR LINE
:368 013224 000704      BR    1$
:369 013226 012604      8$:   MOV   (6)+, R4      :POP STACK INTO R4
:370 013230 012603      MOV   (6)+, R3      :POP STACK INTO R3
:371 013232 012602      MOV   (6)+, R2      :POP STACK INTO R2
:372 013234 012601      MOV   (6)+, R1      :POP STACK INTO R1
:373 013236 012600      MOV   (6)+, R0      :POP STACK INTO R0
:374 013238 000000      RPT

```

```

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,DSWR    ;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 013310      SENDAD =
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          :          $HLT          ERROR TYPEOUT HANDLER
1897
1898          :          ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
1899          :          ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
1900          :          ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
1901          :          ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
1902          :          ;("HLT+3") WILL BE PLACED IN "HLTCTS:" FOR ADITONAL TYPEOUTS.
1903
1904 013326 004737 015006      EMT$:   JSR      PC,      KBDINT
1905 013332 032777 002000 165650      BIT      #SW10,DSWR    ;BELL ON ERROR?
1906 013340 001402          BEQ      1$           ;NO - SKIP
1907 013342 000004 000007      TYPE    .BELL        ;RING BELL
1908 013346 005237 001202 1$:     INC      ERRORS      ;COUNT THE NUMBER OF ERRORS
1909 013352 032777 020000 165630      BIT      #SW13,DSWR    ;SKIP TYPEOUT IF SET
1910 013360 001026          BNE      2$           ;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,HLTCTS ;GET HLT ARGUMENT
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      2$:     JSR      #0,ERRORS  ;GO TO USER ERROR ROUTINE
1919 013436 005777 165546      TST     #SWR          ;HALT "" ERROR
1920 013442 100001          BFL      .+4         ;SKIP IF CONTINUE
1921 013444 000000          HALT          ;HALT ON ERROR!
1922 013446 004737 015006      JSR      PC,KBDINT
1923 013452 000002          RTI          ;RETURN
1924
1925 013454 000000      HLTCTS: 0          ;HLT ARGUMENT
1926 013456 000000      HLTADR: 0          ;LAST HLT INSTRUCTION EXECUTED
1927
1928 013460 042737 007700 013502      ERRORS: BIC      #7700,2$
1929 013466 105337 013454      1$:     DECB    HLTCTS$
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	IS	
1936	013516	000207		3\$:	RTS	PC	

```

1937
1938           ;SUBROUTINE TO SAVE INPUT AS OCTAL NUMBER
1939
1940 013520 012737 000001 014012 READIN: MOV    #1,INHRE
1941 013526 004737 013656          JSR    PC,   READS   ;GO READ TTY UNTIL CR
1942 013532 005037 014012          CLR    INHRE
1943 013536 010146          MOV    R1,-(6)      ;PUSH R1 ON STACK
1944 013540 010246          MOV    R2,-(6)      ;PUSH R2 ON STACK
1945 013542 010346          MOV    R3,-(6)      ;PUSH R3 ON STACK
1946 013544 012501          MOV    (R5)+,R1
1947 013546 012737 000020 014704 MOV    #20,CNT
1948 013554 012702 014014          MOV    #INPUT,R2
1949 013550 122712 000120          CMPB  #120,(R2)    ;CHECK FOR "P"
1950 013564 001425          BEQ   3$
1951 013566 005011          CLP   (R1)
1952 013570 112203          1$:  MOVB  (R2)+,R3
1953 013572 120327 000015          CMPB  R3,#15
1954 013576 001420          BEQ   3$           ;BRANCH WHEN DONE
1955 013600 162703 000060          SUB   #60,R3
1956 013604 032703 177770          BIT   #177770,R3
1957 013610 001013          BNE   3$           ;BRANCH IF BAD DATA
1958 013612 006311          ASL   (R1)
1959 013614 103410          BCS   2$
1960 013616 006311          ASL   (R1)
1961 013620 103406          BCS   2$
1962 013622 006311          ASL   (R1)
1963 013624 103404          BCS   2$
1964 013626 050311          BIS   R3,(R1)
1965 013630 005337 014704          DEC   CNT
1966 013634 000755          BR   1$
1967 013636 000244          2$:  CLZ
1968 013640 013737 177776 013664 3$:  MOV    @#PS, PSTEMP ;MAKE SURE Z-BIT IS CLR
1969 013646 012503          MOV    (6)+,R3     ;SAVE CONDITION CODES
1970 013650 012502          MOV    (6)+,R2     ;POP STACK INTO R3
1971 013652 012601          MOV    (6)+,R1     ;POP STACK INTO R2
1972 013654 013737 013664 177776 MOV    PSTEMP,@#PS ;POP STACK INTO R1
1973 013662 000205          RTS   R5           ;RESTORE CONDITION CODES
1974
1975 013664 000000          PSTEMP 0           ;TEMPORARY STORAGE FOR PS
1976
1977 013666 010346          READS: MOV    R3,-(6) ;SAVE R3
1978 013670 012703 014014          1$:  MOV    #INPUT,R3   ;GET ADDRESS
1979 013674 022703 014034          2$:  CMP    #INPUT+20,R3 ;BUFFER FULL?
1980 013700 001415          BEQ   4$           ;YES - TYPE "?"
1981 013702 105737 177560          TSTB  @#177560    ;WAIT FOR
1982 013706 100375          BPL   -4           ;A CHARACTER
1983 013710 113713 177562          MOVB  @#177562,(3) ;GET CHARACTER
1984 013714 142713 000200          BICB  #200,(3)    ;GET RID OF JUNK
1985 013720 122713 000177          CMPB  #177,(3)   ;IS IT A RUBOUT
1986 013724 001403          BEQ   4$           ;SKIP IF NOT
1987 013726 122713 000025          CMPB  #25,(3)
1988 013732 001006          BNE   3$
1989 013734          4$:
1990 013734 000004 013740          TYPE  .+2         ;.ASCIZ "?"(15)<12\=""
1991 013746 000750          BR   1$           ;ZAP THE BUFFER AND LOOP
1992 013750 111337 014622          3$:  MOVB  (3)..TYPE   ;SET UP FOR TYPING

```

1993	013754	000004	014622		TYPE	..TYPE	:ECHO IT
1994	013760	122723	000015		CMPS	#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	105062	177777	5\$:	CLRB	-1(3)	:ZAP RETURN (THE 15)
0000	014002	000004	000012	6\$:	TYPE	12	:TYPE A LINE FEED
0001	014006	012602			MOV	(6)+.R3	:RESTORE R3
0002	014010	000207			RTS	PC	:RETURN
0003							
0004							
0005							
0006							
0007							
0008							
0009							
0010							
0011							
0012							
0013							
0014							
0015							
0016							
0017							
0018							
0019							
0020							
0021							
0022							
0023							
0024							
0025							
0026							
0027							
0028							
0029							
0030							
0031							
0032							
0033							
0034							
0035							
0036							
0037							
0038							
0039							
0040							
0041							
0042							
0043							
0044							
0045							
0046							
0047							
0048							
0049							
0050							
0051							
0052							
0053							
0054							
0055							
0056							
0057							
0058							
0059							
0060							
0061							
0062							
0063							
0064							
0065							
0066							
0067							
0068							
0069							
0070							
0071							
0072							
0073							
0074							
0075							
0076							
0077							
0078							
0079							
0080							
0081							
0082							
0083							
0084							
0085							
0086							
0087							
0088							
0089							
0090							
0091							
0092							
0093							
0094							
0095							
0096							
0097							
0098							
0099							
0100							
0101							
0102							
0103							
0104							
0105							
0106							
0107							
0108							
0109							
0110							
0111							
0112							
0113							
0114							
0115							
0116							
0117							
0118							
0119							
0120							
0121							
0122							
0123							
0124							
0125							
0126							
0127							
0128							
0129							
0130							
0131							
0132							
0133							
0134							
0135							
0136							
0137							
0138							
0139							
0140							
0141							
0142							
0143							
0144							
0145							
0146							
0147							
0148							
0149							
0150							
0151							
0152							
0153							
0154							
0155							
0156							
0157							
0158							
0159							
0160							
0161							
0162							
0163							
0164							
0165							
0166							
0167							
0168							
0169							
0170							
0171							
0172							
0173							
0174							
0175							
0176							
0177							
0178							
0179							
0180							
0181							
0182							
0183							
0184							
0185							
0186							
0187							
0188							
0189							
0190							
0191							
0192							
0193							
0194							
0195							
0196							
0197							
0198							
0199							
0200							

:TTY INPUT AREA

2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046

014054 012737 170101 014222  
014062 000411  
014064 112737 000001 014222  
014072 000402  
014074 005037 014222  
014100 112737 177772 014223  
014106 010446  
014110 012704 014224  
014114 105014  
014116 000411  
014120 105014  
014122 032737 000100 014222  
014130 001004  
014132 006105  
014134 106114  
014136 006105  
014140 106114  
014142 006105  
014144 106114  
014146 105714  
014150 001402  
014152 105237 014222  
014156 105737 014222  
014162 001402  
014164 152724 000060  
014170 105237 014223  
014174 001351  
014176 022704 014224  
014202 001002  
014204 112724 000060  
014210 105014  
014212 000004 014224  
014216 012604  
014220 000207  
014222 000012

: \$OCTAL OCTAL TYPEOUT ROUTINE

: THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE  
: ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE  
: THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.

BITYPS: MOV #170101,.PR ;SET BIT FLAG ANS 16. CHARACTER COUNT  
BR .PTIT ;NOW TYPE IT IN BIT FORM  
PRINTR: MOVB #1,.PR ;SET ZERO FILL SWITCH  
BR .+6 ;SKIP  
PRINTS: CLR .PR ;SUPPRESS LEADING ZERO'S  
MOVB #-6,.PR+1 ;SET COUNT  
.PTIT: MOV R4,-(6) ;SAVE R4  
MOV #.PR+2,R4 ;SET POINTER TO FIRST ASCII CHAR.  
CLRB (4) ;CLEAR FIRST BYTE  
BR .PRF ;ROTATE FIRST BIT  
.PRL: CLRB (4) ;CLEAR BYTE OF CHARACTER  
BIT #100,.PR ;BIT TYPING MODE?  
BNE .PRF ;YES - SKIP 2 ROTATES  
ROL TTY ;ROTATE BIT INTO C  
ROLB (4) ;PACK IT  
ROL TTY ;ROTATE BIT INTO C  
ROLB (4) ;PACK IT  
.PRF: ROL TTY ;ROTATE BIT INTO C  
ROLB (4) ;PACK IT  
TSTB (4) ;IS IT ZERO?  
BEQ .+6 ;SKIP INC  
INCB .PR ;SET FILL SWITCH  
TSTB .PR ;CHECK FILL SWITCH  
BEQ .+6 ;SKIP BITSET  
BISB #'0,(4)+ ;MAKE INTO ASCII CHAR  
INCB .PR+1 ;INC COUNT  
BNE .PRL ;REPEAT  
CMP #.PR+2,R4 ;EMPTY BUFFER?  
BNE .+6 ;SKIP IF NOT  
MOVB #'0,(4)+ ;LOAD 1 ZERO  
CLRB (4) ;NULL TERMINATOR  
TYPE .PR+2 ;TYPE IT  
MOV (6)+,R4 ;RESTORE R4  
RTS PC ;RETURN  
.PR: .BLKW 12 ;COUNT, SWITCH, AND OUTPUT BUFFER



```

2047 014246 012777 014374 000126 PDOWN$: MOV #ILLUP, @PUVECS ;SET FOR FAST UP
2048 014254 012777 000340 000122 MOV #340, @PUVECS+2 ;PRIO:7
2049 014262 010046 MOV R0, -(6) ;PUSH R0 ON STACK
2050 014264 010146 MOV R1, -(6) ;PUSH R1 ON STACK
2051 014266 010246 MOV R2, -(6) ;PUSH R2 ON STACK
2052 014270 010346 MOV R3, -(6) ;PUSH R3 ON STACK
2053 014272 010446 MOV R4, -(6) ;PUSH R4 ON STACK
2054 014274 010546 MOV R5, -(6) ;PUSH R5 ON STACK
2055 014276 010637 014400 MOV SP, .SAVR6 ;SAVE SP
2056 014302 012777 014312 000072 MOV #PUPS, @PUVECS ;SET UP VECTOR
2057 014310 000000 HALT ;WAIT FOR PF
2058
2059 014312 013706 014400 PUPS: MOV .SAVR6, SP ;GET SP
2060 014316 005001 CLR R1 ;WAIT LOOP FOR THE TTY
2061 014320 005201 1$: 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 PUVECS: 24, 26 ;POWER UP VECTOR
2079
2080
2081 014406 000002 YESRT: RTI ;RETURN FROM TRACE TRAP

```

```

2082
2083
2084
2085
2086
2087
2088 014410 022716 001000
2089 014414 002440
2090 014416 162716 000004
2091 014422 000004 014426
2092 014426 005015 047125 054105
2093 014434 042520 052103 042105
2094 014442 044440 052116 051105
2095 014450 050125 020124 047524
2096 014456 000040
2097 014460 012605
2098 014462 004737 014074
2099 014466 005726
2100 014470 000004 014474
2101 014474 043040 047522 020115
2102 014502 000
2103 014504
2104 014504 011605
2105 014506 004737 014074
2106 014512 000000
2107 014514 000002
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117 014516 010546
2118 014520 017605 000002
2119 014524 032705 177400
2120 014530 001004
2121 014532 010537 014622
2122 014536 012705 014622
2123 014542 105715
2124 014544 001406
2125 014546 112537 177566
2126 014552 105737 177566
2127 014556 100375
2128 014560 000770
2129 014562 017646 000002
2130 014566 062766 000002 000004
2131 014574 022666 000002
2132 014600 001006
2133 014602 062705 000002
2134 014606 042705 000001
2135 014612 010566 000002
2136 014616 012605
2137 014620 000002

```

```

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

```

```

IOTRAP: CMP #1000, (SP) ;CHECK RETURN ADDRESS
        BLT IOTS ;BRANCH IF TYPE COMMAND
        SUB #4, (SP) ;GET VECTOR ADDRESS
        TYPE +2 ;TYPE MESSAGE
        .ASCIZ <15><12>"UNEXPECTED INTERRUPT TO "

        MOV (SP)+, TTY ;TYPE (SP)+ IN OCTAL
        JSR PC, PRINTS ;AND SUPPRESS LEADING ZERO'S
        TST (SP)+ ;POP STACK
        TYPE +2 ;TYPE MESSAGE
        .ASCIZ " FROM "

        .EVEN
        MOV (SP), TTY ;TYPE (SP) IN OCTAL
        JSR PC, PRINTS ;AND SUPPRESS LEADING ZERO'S
        HALT ;FATAL ERROR
        RTI ;CONTINUE IF DESIRED

```

```

; STYPE MESSAGE TIMEOUT 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.

```

```

IOTS: MOV TTY, -(6) ;SAVE TTY
      MOV @2(6), TTY ;GET ADDRESS TO BE TYPED
      BIT #177400, TTY ;IS IT A TYPEN?
      BNE IS ;NO
      MOV TTY, TYPE ;GET THE CHARACTER
      MOV #TYPE, TTY ;FUDGE THE ADDRESS
IS: TSTB (TTY) ;TERMINATOR?
    BEQ 2$ ;GET OUT IF SO
    MOVB (TTY)+, @#177566 ;LOAD AND TYPE THE CHARACTER
    TSTB @#177566 ;IS THE PRINTER READY
    BPL -4 ;WAIT UNTIL IT IS
    BR IS ;GET THE NEXT CHARACTER
=3: MOV @2(6), -(6) ;GET ADDRESS TO BE TYPED
    ADD #2, 4(6) ;ADD 2 TO THE ADDRESS
    CMP (6)+, 2(6) ;IS IT .+2?
    BNE 3$ ;NO
    ADD #2, TTY ;ADD 2 TO THE ADDRESS
    BIC #1, TTY ;BACK UP TO AN EVEN BYTE
    MOV TTY, 2(6) ;RESTORE ADDRESS
3$: MOV (6)+, TTY ;RESTORE TTY
    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 RS,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 TMP1,2SWR
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 #-1,2SWR
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	030512	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 =	009007	459*	1882	1907															
BINARY	007760	1045	1061*																
BITYPS	014054	2012*																	
BIT0 =	000901	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	011704	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*															
EMTS	013326	561	1904*																
END	015360	1051	1053	1055	1057	1059	1604	2227*											
ERRORS	001202	499*	570*	1908*	1925														
ERRORS	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*													
HLTCTS	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*														
ICTRAP	014410	557	2088*																
IOTS	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					













G05

MAINDEC-11-DZCJB-D-D DJ11 EXERCISER AND ON-LINE TESTS  
DZDJBO.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

MACY11 27(732) 21-SEP-76 13:54 PAGE 60

.PTIT	014106	2013	2018*			
.SAVR6	J14400	2055*	2059	2077*		
.TBIT	013324	:894*				
.TYPE	J1462E	1992*	1993	2121*	212E	2138*



MAINDEC-11-DZDJ8-D-0 DJ11 EXERCISER AND ON-LINE TESTS  
DZDJ8D.P11 CROSS REFERENCE TABLE -- MACRO NAMES

SSWRDF	1*	502
SSWRRR	1*	2161
STRAP	1*	
STYPE	1*	2109
SURAT	1*	
.SCUP	1*	
.SCOPE	1*	

MAINDEC-11-DZDJ8-D-D DJ11 EXERCISER AND ON-LINE TESTS  
 DZDJ8D.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

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	1825	1878	1931	2130	2133				
ASL	673	674	675	676	691	744	811	817	818	828	944	1018	1019	1279	1389
	1489	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											
BOS	590	1959	1961	1963											
BEG	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	1912	1853	1880	1905
	1909	1956	2023	2119											
BITB	1550	1826													
BLC	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	697	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	1956	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	2155	2169	2175			
CMPB	998	1949	1959	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	1909
	2061														
INCB	2033	2037													
ICT	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	86	878	880	892
	984	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

# K05

MAINDEC-11-DZDJ8-D-D  
DZDJ80.P11

DJ11 EXERCISER AND ON-LINE TESTS  
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 21-SEP-76 13:54 PAGE 66

MOV	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				
	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	38	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	1639	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								
MOV8	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	1989	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	1398	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

# LOS

MAINDEC-11-DZJJB-D-D  
DZJBD.P11

DJ11 EXERCISER AND ON-LINE TESTS  
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 21-SEP-76 13:54 PAGE F7

	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	1297	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	1890	1882	1884	1893	1909	1911	1919	1924	1980	2006	2022	2047	2055	2063
	2072	2073	2123	2138											
.EVEN	730	1329	1912	1318	1991	2072	2103	2226							
.IF	460	1880	1881	1884	1892	1904	1909	1918	1923	1979	205	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	206	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	1294
	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299



M05

MAINDEC-11-DZDJ8-D-0 DJ11 EXERCISER AND ON-LINE TESTS  
 DZDJ8D.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 21-SEP-76 13:54 PAGE 68

	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
.PAGE	1912	1918	1937	1977	1991	2006	2047	2072	2082	2109					
.REM	476	1937	2006	2047	2082										
.REPT	1	402													
.SBTTL	492	1062	1285												
	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
.TITLE	2082	2109													
.WORD	385														
	717	731	869	871	873	875	877	879	881	883	885	887	889	891	933
	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

\* ,DZDJ8D.SEG/SOL/CRF=DZDJ8D.MAC,DZDJ8D.P11  
 RUN-TIME: 19 26 4 SECONDS  
 RUN-TIME RATIO: 238/50=4.6  
 CORE USED: 21K (41 PAGES)

