

```

762      ;          .TITLE 'CENTRAL INPUT/OUTPUT (CIO) 2-7-79'
763      ;          UPDATED BY AL MILLER 3-9-79
764      0030      ASCZER      =      '0          ;ASCII ZERO
765      003A      COLON      =      $3A          ;ASCII COLON
766      0098      EOL      =      $9B          ;END OF RECORD
767      ;.PAGE
768      ;
769      ; CIO JUMP VECTOR FOR USERS
770      ;          *=CIOV
771      E456      4C C4 E4      JMP      CIO          ;GO TO CIO
772      ;
773      ; CIO INIT JUMP VECTOR FOR POWER UP
774      ;          *=CIOINV
775      E46E      4C A6 E4      JMP      CIOINT      ;GO TO INIT
776      ;
777      ;
778      ; ERROR ROUTINE ADDRESS EQUATE
779      ; ERRTNH =ERRTN/256 "MOVED TO LINE 788"
780      ; ERRTNL =-ERRTNH*256+ERRTN "MOVED TO LINE 789"
781      ;
782      ;
783      ;          *=CIOORG
784      ;
785      ; CIO INITIALIZATION (CALLED BY MONITOR AT POWER UP)
786      E4A6      A2 00      CIOINT:      LDX      #0
787      E4A8      A9 FF      CIOI1:      LDA      #IOCFRE      ;SET ALL IOCB'S TO FREE
788      E4AA      9D 40 03      STA      ICHID,X      ;BY SETTING HANDLER ID'S=$FF
789      E4AD      A9 C0      LDA      #ERRTNL
790      E4AF      9D 46 03      STA      ICPTL,X      ;POINT PUT TO ERROR ROUTINE
791      E4B2      A9 E4      LDA      #ERRTNH
792      E4B4      9D 47 03      STA      ICPH,X
793      E4B7      8A      TXA
794      E4B8      18      CLC
795      E4B9      69 10      ADC      #IOCBSZ      ;BUMP INDEX BY SIZE
796      E48B      AA      TAX
797      E4BC      C9 80      CMP      #MAXIOC      ;DONE?
798      E4BE      90 E8      BCC      CIOI1      ;NO
799      E4C0      60      RTS          ;YES, RETURN
800      ;

```

```

801      ; ERROR ROUTINE FOR ILLEGAL PUT
802      ERRTN      =      *-1
803      ERRTNH     =      ERRTN/256
804      ERRTNL     =      (-ERRTNH)*256+ERRTN
805      E4C1      A0 85      LDY      #NOTOPN      ;IOCB NOT OPEN
806      E4C3      60      RTS
807      ; .PAGE
808      ;
809      ; CIO LOCAL RAM (USES SPARE BYTES IN ZERO PAGE IOCB)
810      002C      ENTVEC     =      ICSPRZ
811      ;
812      ; CIO MAIN ROUTINE
813      ;
814      ; CIO INTERFACES BETWEEN USER AND INPUT/OUTPUT DE
815      E4C4      85 2F      CIO:      STA      CIOCHR      ;SAVE POSSIBLE OUTPUT CHARACTER
816      E4C6      86 2E      STX      ICIDNO      ;SAVE IOCB NUMBER * N
817      ;
818      ; CHECK FOR LEGAL IOCB
819      I819      E4C8      8A      TXA
820      E4C9      29 0F      AND      #$F      ;IS IOCB MULTIPLE OF 16?
821      E4CB      D0 04      BNE      CIERR1      ;NO, ERROR
822      E4CD      E0 80      CPX      #MAXIOC      ;IS INDEX TOO LARGE?
823      E4CF      90 05      BCC      IOC1      ;NO
824      ;
825      ; INVALID IOCB NUMBER -- RETURN ERROR
826      E4D1      A0 86      CIERR1:  LDY      #BADIOC      ;ERROR CODE
827      E4D3      4C 1B E6      JMP      CIRTN1      ;RETURN
828      ;
829      ; MOVE USER IOCB TO ZERO PAGE
830      E4D6      A0 00      IOC1:      LDY      #0
831      E4D8      BD 40 03      IOC1A:  LDA      IOCB,X      ;USER IOCB
832      E4DB      99 20 00      STA      IOCBAS,Y      ;TO ZERO PAGE
833      E4DE      E8      INX
834      E4DF      C8      INY
835      E4E0      C0 0C      CPY      #12      ;12 BYTES
836      E4E2      90 F4      BCC      IOC1A
837      ;
838      ; COMPUTE CIO INTERNAL VECTOR FOR COMMAND
839      E4E4      A0 84      LDY      #INVALID      ;ASSUME INVALID CODE

```

```

840 E4E6 A5 22 LDA ICCOMZ ;COMMAND CODE TO INDEX
841 E4EB C9 03 CMP #OPEN ;IS COMMAND LEGAL?
842 E4EA 90 25 BCC CIERR4 ;NO
843 E4EC A8 TAY
844 ;
845 ; MOVE COMMAND TO ZERO BASE FOR INDEX
846 E4ED C0 0E CPY #SPECIL ;IS COMMAND SPECIAL"?
847 E4EF 90 02 BCC IOC2 ;NO
848 E4F1 A0 0E LDY #SPECIL ;YES, SET SPECIAL OFFSET INDEX
849 E4F3 84 17 IOC2: STY I CCOMT ;SAVE COMMAND FOR VECTOR
850 E4F5 B9 C6 E6 LDA COMTAB-3,Y ;GET VECTOR OFFSET FROM TABLE
851 E4F8 F0 0F BEQ CIOPEN ;GO IF OPEN COMMAND
852 E4FA C9 02 CMP #2 ;IS IT CLOSE?
853 E4FC F0 35 BEQ CICLOS ;YES
854 E4FE C9 08 CMP #8 ;IS IT STATUS OR SPECIAL?
855 E500 B0 4C BCS CISTSP ;YES
856 E502 C9 04 CMP #4 ;IS IT READ?
857 E504 F0 63 BEQ CIREAD ;YES
858 E506 4C C9 E5 JMP CIWRIT ;ELSE, MUST BE WRITE
859 ; .PAGE
860 ;
861 ; OPEN COMMAND
862 ;
863 ; FIND DEVICE HANDLER IN HANDLER ADDRESS TABLE
864 E509 A5 20 CIOPEN: LDA ICHIDZ ;GET HANDLER ID
865 E50B C9 FF CMP *IOCFRE ;IS THIS IOCB CLOSED?
866 E500 F0 05 BEQ IDC6 ;YES
867 ;
868 ; ERROR -- IOCB ALREADY OPEN
869 E50F A0 81 CIERR3: LDY *PRVOPN ;ERROR CODE
870 E511 4C 1B E6 CIERR4: JMP CIRTNI ;RETURN
871 ;
872 ; GO FIND DEVICE
873 E514 20 9E E6 IOC6: JSR DEVSRC ;CALL DEVICE SEARCH
874 E517 B0 F8 BCS CIERR4 ;GO IF DEVICE NOT FOUND
875 ;
876 ; DEVICE FOUND. INITIALIZE IOCB FOR OPEN
877 ;
878 ; COMPUTE HANDLER ENTRY POINT

```

```

879 E519 20 3D E6      IOC7:      JSR      COMENT
880 E51C   B0F3          BCS      CIERR4      ;GO IF ERROR IN COMPUTE
881                      ;
882                      ; GO TO HANDLER FOR INITIALIZATION
883 E51E 20 89 E6          JSR      GOHAND      ;USE INDIRECT JUMP
884                      ;
885                      ; STORE PUT BYTE ADDRESS-1 INTO IOCB
886 E521 A9 0B          LDA      *PUTCHR      ;SIMULATE PUT CHARACTER
887 E523 85 17          STA      ICCOMT
888 E525 20 3D E6          JSR      COMENT      ;COMPUTE ENTRY POINT
889 E528 A5 2C          LDA      ICSPRZ      ;MOVE COMPUTED VALUE
890 E52A 85 26          STA      ICPTLZ      ;TO PUT BYTE ADDRESS
891 E52C A5 2D          LDA      ICSPRZ+1
892 E52E 85 27          STA      ICPTHZ
893 E530 4C ID E6          JMP      CIRTN2      ;RETURN TO USER
894                      ; .PAGE
895                      ;
896                      ;
897                      ; CLOSE COMMAND
898 E533 A0 01      CICLOS:      LDY      #SUCCES      ;ASSUME GOOD CLOSE
899 E535 84 23          STY      ICSTAZ
900 E537 20 3D E6          JSR      COMENT      ;COMPUTE HANDLER ENTRY POINT
901 E53A B0 03          BCS      CICL02      ;GO IF ERROR IN COMPUTE
902 E53C 20 89 E6          JSR      GOHAND      ;GO TO HANDLER TO CLOSE DEVICE
903 E53F A9 FF      CICL02:      LDA      #IOCFRE      ;GET IOCB "FREE" VALUE
904 E541 85 20          STA      ICHIDZ      ;SET HANDLER ID
905 E543 A9 E4          LDA      #ERRTNH
906 E545 85 27          STA      ICPTHZ      ;SET PUT BYTE TO POINT TO ERROR
907 E547 A9 C0          LDA      #ERRTNL
908 E549 85 26          STA      ICPTLZ
909 E54B 4C ID E6          JMP      CIRTN2      ;RETURN
910                      ;
911                      ;
912                      ; STATUS AND SPECIAL REQUES TS
913                      ; D0IMPLIED OPEN IF NECESSARY AND GO TO DEVICE
914 E54E A5 20      CISTSP:      LDA      ICHIDZ      ;IS THERE A HANDLER ID?
915 E550 C9 FF          CMP      #IOCFRE
916 E552 D0 05          BNE      CIST1      ;YES
917                      ;

```

```

918      ; IOCB IS FREE, DO IMPLIED OPEN
919      E554 20 9E E6      JSR  DEVSRC      ;FIND DEVICE IN TABLE
920      E557 B0 B8      BCS  CIERR4      ;GO IF ERROR IN COMPUTE
921      ;
922      ; COMPUTE AND GO TO ENTRY P      OINT IN HANDLER
923      E559 20 3D E6      CIST1: JSR  COMENT      ;COMPUTER HANDLER ENTRY VECTOR
924      E55C 20 89 E6      JSR  GOHAND      ;GO TO HANDLER
925      ;
926      ; RESTORE HANDLER INDEX (DO      IMPLIED CLOSE)
927      E55F A6 2E      LDX  ICIDNO      ;IOCB INDEX
928      E561 BD 40 03      LDA  ICHID,X      ;GET ORIGINAL HANDLER ID
929      E564 85 20      STA  ICHIDZ      ;RESTORE ZERO PAGE
930      E566 4C 1D E6      JMP  CIRTN2      ;RETURN
931      ;      .PAGE
932      ;
933      ; READ      -- D0GET COMMANDS
934      E569 A5 22      CIREAD: LDA  ICCOMZ      ;GET COMMAND BYTE
935      E56B 25 2A      AND  ICAX1Z      ;IS THIS READ LEGAL?
936      E56D D0 05      BNE  RCI1A      ;YES
937      ;
938      ; ILLEGAL READ -- IOCB OPENED FOR WRITE ONLY
939      E56F A0 83      LDY  *WRONLY      ;ERROR CODE
940      E571 4C 1B E6      RCI1B: JMP  CIRTN1      ;RETURN
941      ;
942      ;COMPUTE AND      CHECK ENTRY POINT
943      E574 20 3D E6      RCI1A: JSR  COMENT      ;COMPUTE ENTRY POINT
944      E577 B0 F8      BCS  RCI1B      ;GO IF ERROR IN COMPUTE
945      ;
946      ;GET RECORD OR CHARACTERS
947      E579 A5 28      LDA  ICBLLZ
948      E57B 05 29      ORA  ICBLLZ+1      ;IS BUFFER LENGTH ZERO?
949      E57D D0 08      BNE  RC13      ;NO
950      E57F 20 89 E6      JSR  GOHAND
951      E582 85 2F      STA  CIOCHR
952      E584 4C 1D E6      JMP  CIRTN2
953      ;
954      ; LOOP      TO FILL      BUFFER OR END RECORD
955      E587 20 89 E6      RC13: JSR  GOHAND      ;GO TO HANDLER TO GET BYTE
956      E58A 85 2F      STA  CIOCHR      ;SAVE BYTE

```

```

957 E58C 30 35 BMI RC14 ;END TRANSFER IF ERROR
958 E58E A0 00 LDY #0
959 E590 91 24 STA (ICBALZ),Y ;PUT BYTE IN USER BUFFER
960 E592 20 70 E6 JSR INCBFP ;INCREMENT BUFFER POINTER
961 E595 A5 22 LDA ICCOMZ ;GET COMMAND CODE
962 E597 29 02 AND #2 ;IS IT GET RECORD?
963 E599 D0 0C BNE RCI1 ;NO
964 ;
965 ; CHECK FOR EOL ON TEXT RECORDS
966 E598 A5 2F LDA CIOCHR ;GET BYTE
967 E59D C9 9B CMP 11E0L ;IS IT AN EOL?
968 E59F D006 BNE RCI1 ;NO
969 E5A1 20 63 E6 JSR DECBFL ;YES, DECREMENT BUFFER LENGTH
970 E5A4 4C C3 E5 JMP RC14 ;END TRANSFER
971 ;
972 ; CHECK BUFFER FULL
973 E5A7 20 63 E6 RCI1: JSR DECBFL ;DECREMENT BUFFER LENGTH
974 E5AA D0 DB BNE RC13 ;CONTINUE IF NON ZERO
975 ; .PAGE
976 ;
977 ; BUFFER FULL. RECORD NOT ENDED
978 ; DISCARD BYTES UNTIL END OF RECORD
979 E5AC A5 22 RCI2: LDA ICCOMZ ;GET COMMAND BYTE
980 E5AE 29 02 AND *2 ;IS IT GET CHARACTER?
981 E5B0 D0 11 BNE RC14 ;YES. END TRANSFER
982 ;
983 ; LOOP TO WAIT FOR EOL
984 E5B2 20 89 E6 RCI6: JSR GOHAND ;GET BYTE FROM HANDLER
985 E5B5 85 2F STA CIOCHR ;SAVE CHARACTER
986 E5B7 30 0A BMI RCI4 ;GO IF ERROR
987 ;
988 ; TEXT RECORD WAIT FOR EOL
989 E5B9 A5 2F LDA CIOCHR ;GET GOT BYTE
990 E5BB C9 98 CMP #EOL ;IS IT EOL?
991 E5BD D0 F3 BNE RC16 ;NO. CONTINUE
992 ;
993 ; END OF RECORD BUFFER FULL -- SEND TRUNCATED RECORD MESSAGE
994 E5BF A9 89 RCI11: LDA #TRNRCD ;ERROR CODE
995 E5C1 85 23 STA ICSTAZ ;STORE IN IOCB

```

```

996      ;
997      ; TRANSFER DONE
998      E5C3  20 77 E6      RCI4:      JSR      SUBBFL      ;SET FINAL BUFFER LENGTH
999      E5C6  4C 1D E6      JMP      CIRTN2      ;RETURN
1000     ; .PAGE
1001     ;
1002     ; WRITE -- DOPUT COMMANDS      .
1003      E5C9  A5 22      CIWRIT:      LDA      ICCOMZ      ;GET COMMAND BYTE
1004      E5CB  25 2A      AND      ICAX1Z      ;IS THIS WRITE LEGAL?
1005      E5CD  D0 05      BNE      WCI1A      ;YES
1006     ;
1007     ; ILLEGAL WRITE -- DEVICE      OPENED FOR READ ONLY
1008      E5CF  A0 87      LDY      #RDONLY      ;ERROR CODE
1009      E5D1  4C 1B E6      WCI1B:      JMP      CIRTN1      ;RETURN
1010
1011     ; COMPUTE AND CHECK ENTRY      POINT
1012      E5D4  20 3D E6      WCI1A:      JSR      COMENT      ;COMPUTE HANDLER ENTRY POINT
1013      E5D7  B0 F8      BCS      WCI1B      ;GO IF ERROR IN COMPUTE
1014     ;
1015     ; PUT RECORD OR CHARACTERS
1016      E5D9  A5 28      LDA      ICBLLZ
1017      E5DB  05 29      ORA      ICBLLZ+1      ;IS BUFFER LENGTH ZERO?
1018      E5DD  D0 06      BNE      WCI3      ;NO
1019      E5DF  A5 2F      LDA      CIOCHR      ;GET CHARACTER
1020      E5E1  E6 28      INC      ICBLLZ      ;SET BUFFER LENGTH=i
1021      E5E3  D0 06      BNE      WCI4      ;THEN JUST TRANSFER ONE BYTE
1022     ;
1023     ; LOOP TO TRANSFER BYTES FROM BUFFER TO HANDLER
1024      E5E5  A0 00      WCI3:      LDY      #0
1025      E5E7  B1 24      LDA      (ICBALZ),Y      ;GET BYTE FROM BUFFER
1026      E5E9  85 2F      STA      CIOCHR      ;SAVE
1027      E5EB  20 89 E6      WCI4:      JSR      GOHAND      ;GO PUT BYTE
1028      E5EE  30 25      BMI      WCI5      ;END IF ERROR
1029      E5F0  20 70 E6      JSR      INCBFP      ;INCREMENT BUFFER POINTER
1030     ;
1031     ; CHECK FOR TEXT RECORD
1032      E5F3  A5 22      LDA      ICCOMZ      ;GET COMMAND BYTE
1033      E5F5  29 02      AND      #2      ;IS IT PUT RECORD?
1034      E5F7  D0 0C      BNE      WCI1      ;NO

```

```

1035      ;
1036      ; TEXT RECORD -- CHECK FOR EOL TRANSFER
1037      E5F9  A5 2F      LDA  CIOCHR      ;GET LAST CHARACTER
1038      E5FB  C9 9B      CMP  #EOL      ;IS IT AN EOL?
1039      E5FD  D0 06      BNE  WCI1      ;NO
1040      E5FF  20 63 E6    JSR  DECBFL      ;DECREMENT BUFFER LENGTH
1041      E602  4C 15 E6    JMP  WCI5      ;END TRANSFER
1042      ;
1043      ; CHECK FOR BUFFER EMPTY
1044      E605  20 63 E6    WCI1: JSR  DECBFL      ;DECREMENT BUFFER LENGTH
1045      E608  D0 DB      BNE  WCI3      ;CONTINUE IF NON ZERO
1046      ;.PAGE
1047      ;
1048      ; BUFFER EMPTY      RECORD NOT  FILLED
1049      ; CHECK TYPE OF    TRANSFER
1050      E60A  A5 22      WCI2: LDA  ICCOMZ      ;GET COMMAND CODE
1051      E60C  29 02      AND  IR2      ;IS IT PUT CHARACTER?
1052      E60E  D0 05      BNE  WCI5      ;YES END TRANSFER
1053      ;
1054      ; PUT RECORD (TEXT) BUFFER      EMPTY. SEND EOL
1055      E610  A9 9B      LDA  #EOL
1056      E612  20 89 E6    JSR  GOHAND      ;GO TO HANDLER
1057      ;
1058      ; END PUT TRANSFER
1059      E615  20 77 E6    WCI5: JSR  SUBBFL      ;SET ACTUAL PUT BUFFER LENGTH
1060      E618  4C 1D E6    JMP  CIRTN2      ;RETURN
1061      ; .PAGE
1062      ;
1063      ; CIO RETURNS
1064      ; RETURNS WITH      Y=STATUS
1065      E61B  84 23      CIRTN1: STY  ICSTAZ      ;SAVE STATUS
1066      ;
1067      ; RETURNS WITH STATUS STORED IN ICSTAZ
1068      ; MOVE IOCB IN ZERO PAGE BACK TO USER AREA
1069      E61D  A4 2E      CIRTN2: LDY  ICIDNO      ;GET IOCB INDEX
1070      E61F  B9 44 03    LDA  ICBAL,Y
1071      E622  85 24      STA  ICBALZ      ;RESTORE USER BUFFER POINTER
1072      E624  B9 45 03    LDA  ICBAH,Y
1073      E627  85 25      STA  ICBAHZ

```



```

1074 E629 A2 00          LDX    #0          ;LOOP COUNT AND INDEX
1075 E62B 85 20          LDA    IOCBAS~X    ;ZERO PAGE
1076 E62D 99 40 03      CIRT3: STA    IOCB,Y    ;TO USER AREA
1077 E630 E8            INX
1078 E631 C8            INY
1079 E632 E0 0C          CPX    #12          ;12 BYTES
1080 E634 90 F5          BCC    CIRT3
1081                      ;
1082                      ; RESTORE A. X.    & Y
1083 E636 A5 2F          LDA    CIOCHR        ;GET LAST CHARACTER
1084 E638 A6 2E          LDX    ICIDNO        ;IOCB INDEX
1085 E63A A4 23          LDY    ICSTAZ        ;GET STATUS AND SET FLAGS
1086 E63C 60            RTS                    ;RETURN TO USER
1087                      ; .PAGE
1088                      ;
1089                      ;
1090                      : CIO SUBROUTINES
1091                      ;
1092                      ; COMENT -- CHECK AND COMPUTE HANDLER ENTRY POINT
1093 E63D A4 20          COMENT: LDY    ICHIDZ    ;GET HANDLER INDEX
1094 E63F C0 22          CPY    #MAXDEV+1    ;IS IT A LEGAL INDEX?
1095 E641 90 04          BCC    COM1          ;YES
1096                      ;
1097                      ; ILLEGAL HANDLER INDEX MEANS DEVICE NOT OPEN FOR OPERATION
1098 E643 A0 85          LDY    #NDTOPN        ;ERROR CODE
1099 E645 B0 1B          BCS    COM2          ;RETURN
1100                      ;
1101                      ; USE HANDLER ADDRESS TABLE AND COMMAND TABLE TO GET VECTOR
1102 E647 B9 1B 03      COM1:  LDA    HATABS+1,Y  ;GET LOW BYTE OF ADDRESS
1103 E64A 85 2C          STA    ICSPRZ        ;AND SAVE IN POINTER
1104 E64C B9 1C 03      LDA    HATABS+2,Y  ;GET HI BYTE OF ADDRESS
1105 E64F 85 2D          STA    ICSPRZ+1
1106 E651 A4 17          LDY    ICCOMT        ;GET COMMAND CODE
1107 E653 B9 C6 E6      LDA    COMTAB-3,Y  ;GET COMMAND OFFSET
1108 E656 A8            TAY
1109 E657 B1 2C          LDA    (ICSPRZ),Y  ;GET LOW BYTE OF VECTOR FROM
1110 E659 AA            TAX                    ;HANDLER ITSELF AND SAVE
1111 E65A C8            INY
1112 E65B B1 2C          LDA    (ICSPRZ),Y  ;GET HI BYTE OF VECTOR

```

```

1113 E65D 85 2D          STA  ICSPRZ+i
1114 E65F 86 2C          STX  ICSPRZ      ;SET LO BYTE
1115 E661 18             CLC              ;SHOW NO ERROR
1116 E662 60             COM2:  RTS
1117                      ;
1118                      ;
1119                      ;DECBFL -- DECREMENT BUFFER LENGTH DOUBLE BYTE
1120                      ; Z FLAG = 0 ON RETURN IF LENGTH = 0 AFTER DECREMENT
1121 E663 C6 28          DECBFL:  DEC  ICBLLZ      ;DECREMENT LOW BYTE
1122 E665 A5 2B          LDA  ICBLLZ      ;CHECK IT
1123 E667 C9 FF          CMP  #$FF        ;DID IT GO BELOW?
1124 E669 D0 02          BNE  DECBFI      ;NO
1125 E66B C6 29          DEC  ICBLLZ+1    ;DECREMENT HI BYTE
1126 E66D 05 29          DECBFI:  ORA  ICBLLZ+1    ;SET Z IF BOTH ARE ZERO
1127 E66F 60             RTS
1128                      ;
1129                      ;
1130                      ; INCBFP -- INCREMENT WORKING BUFFER POINTER
1131 E670 E6 24          INCBFP:  INC  ICBALZ      ;BUMP LOW BYTE
1132 E672 D0 02          BNE  INCBF1      ;GO IF NOT ZERO
1133 E674 E6 25          INC  ICBALZ+1    ;ELSE, BUMP HI BYTE
1134 E676 60             INCBFI:  RTS
1135                      ;
1136                      ;
1137                      ; SUBBFL -- SET BUFFER LENGT  H = BUFFER LENGTH - WORKING BYTE COUNT
1138 E677 A6 2E          SUBBFL:  LDX  ICIDNO      ;GET IOCB INDEX
1139 E679 38             SEC
1140 E67A BD 48 03          LDA  ICBLL,X      ;GET LOW BYTE OF INITIAL LENGTH
1141 E67D E5 28          SBC  ICBLLZ      ;SUBTRACT FINAL LOW BYTE
1142 E67F 85 28          STA  ICBLLZ      ;AND SAVE BACK
1143 E681 BD 49 03          LDA  ICBLH,X      ;GET HI BYTE
1144 E684 E5 29          SBC  ICBLLZ+1
1145 E686 85 29          STA  ICBLHZ
1146 E688 60             RTS
1147                      ;
1148                      ;
1149                      ; GOHAND - GO INDIRECT TO A DEVICE HANDLER
1150                      ; Y- STATUS ON  RETURN. N FLAG=1 IF ERROR ON RETURN
1151 E689 A0 92          GOHAND:  LDY  #FNCNOT    ;PREPARE NO FUNCTION STATUS FOR HANDLER RTS

```

```

1152 E68B 20 93 E6      JSR   CIJUMP      ;USE THE INDIRECT JUMP
1153 E68E 84 23      STY   ICSTAZ      ;SAVE STATUS
1154 E690 C0 00      CPY   #0        ;AND SET N FLAG
1155 E692 60          RTS
1156                  ;
1157                  ; INDIRECT JUMP TO HANDLER BY PAUL'S METHOD
1158 E693 AA          CIJUMP: TAX      ;SAVE A
1159 E694 A5 2D      LDA   ICSPRZ+1    ;GET JUMP ADDRESS HI BYTE
1160 E696 48          PHA          ;PUT ON STACK
1161 E697 A5 2C      LDA   ICSPRZ      ;GET JUMP ADDRESS LO BYTE
1162 E699 48          PHA          ;PUT ON STACK
1163 E69A 8A          TXA          ;RESTORE A
1164 E69B A6 2E      LDX   ICIDNO      ;GET IOCB INDEX
1165 E69D 60          RTS            ;GO TO HANDLER INDIRECTLY
1166                  ; .PAGE
1167                  ;
1168                  ; DEVSRC -- DEVICE SEARCH FIND DEVICE IN HANDLER ADDRESS TABLE
1169                  ;
1170                  ; LOOP TO FIND DEVICE
1171 E69E A0 00      DEVSRC: LDY   #0
1172 E6A0 B1 24      LDA   (ICBALZ),Y    ;GET DEVICE NAME FROM USER
1173 E6A2 F0 0C      BEQ   CIERR2
1174 E6A4 A0 21      LDY   #MAXDEV      ;INITIAL COMPARE INDEX
1175 E6A6 D9 1A 03    DEVS1: CMP   HATABS,Y    ;IS THIS THE DEVICE?
1176 E6A9 F0 0A      BEQ   DEVS2        ;YES
1177 E6AB 88          DEY
1178 E6AC 88          DEY              ;ELSE. POINT TO NEXT DEVICE NAME
1179 E6AD 88          DEY
1180 E6AE 10 F6      BPL   DEVS1        ;CONTINUE FOR ALL DEVICES
1181
1182                  ; NO DEVICE FOUND, DECLARE NON-EXISTENT DEVICE ERROR
1183 E6B0 A0 82      CIERR2: LDY   #NONDEV    ;ERROR CODE
1184 E6B2 38          SEC              ;SHOW ERROR
1185 E6B3 B0 13      BCS   DEVS4        ;AND RETURN
1186                  ;
1187                  ; FOUND DEVICE. SET ICHID, ICDNO, AND INIT DEVICE
1188 E6B5 98          DEVS2: TYA
1189 E6B6 85 20      STA   ICHIDZ      ;SAVE HANDLER INDEX
1190 E6B8 38          SEC

```

```

1191 E689 A0 01 LDY #1
1192 E6BB B1 24 LDA (ICBALZ),Y ;GET DEVICE NUMBER (DRIVE NUMBER)
1193 E6BD E9 30 SBC #ASCZER ;SUBTRACT ASCII ZERO
1194 E6BF C9 0A CMP #A ;IS NUMBER IN RANGE?
1195 E6C1 90 02 BCC DEVS3 ;YES
1196 E6C3 A9 01 LDA IR1 ;NO, DEFAULT TO ONE
1197 E6C5 85 21 DEVS3: STA ICDNOZ ;SAVE DEVICE NUMBER
1198 E6C7 18 CLC ;SHOW NO ERROR
1199 ;
1200 ; RETURN
1201 E6C8 60 DEVS4: RTS
1202 ; .PAGE
1203 ;
1204 ;
1205 ; CIO ROM TABLES
1206 ;
1207 ; COMMAND TABLE
1208 ; MAPS EACH COMMAND TO OFFSET FOR APPROPRIATE VECTOR IN HANDLER
1209 E6C9 00 04 04 04 COMTAB: .BYTE 0, 4, 4, 4, 4, 6, 6, 6, 6, 2, 8, 10
1210 E6CD 04 06 06 06
1211 E6D1 06 02 08 0A
1212 022E LENGTH = *-CIOINT
1213 E6D5 CRNTP1 =*
1214 *= $14
1215 0014 00 CIOSPR: .BYTE INTORG-CRNTP1 ;^GCIOL IS TOO LONG
1216 ;
1217 ; .TITLE 'INTERRUPT HANDLER'
1218 ; LIVES ON DKI:INTHV.SRC
1219 0006 SRTIM2 = 6 ; SECOND REPEAT INTERVAL
1220 ;
1221 ; THIS IS TO MAKE DOS 2 WORK WHICH USED AN ABSOLUTE ADDRESS
1222 ;
1223 *= $E912
1224 E912 4C ED E8 JMP SETVBL
1225 *=SETVBV
1226 E45C 4C ED E8 JMP SETVBL
1227 E45F 4C AE E7 JMP SYSVBL
1228 E462 4C 05 E9 JMP XITVBL
1229 *=INTINV

```

ERR LINE ADDR B1 B2 B3 B4

CENTRAL INPUT/OUTPUT (CIO) 2-7-79

Page 34

```

1230 E46B 4C D5 E6      JMP      IHINIT
1231                      ;
1232                      *=VCTABL+INTABS-VDSLST
1233
1234 E480 90 E7          .WORD SYRTI      ;VDSLST
1235 E482 8F E7          .WORD SYIRQB     ;VPRCED
1236 E484 8F E7          .WORD SYIRQB     ;VINTER
1237 E486 8F E7          .WORD SYIRQB     ;VBREAK
1238
1239 E488                .RES      8
1240 E490 8F E7          .WORD SYIRQB     ;VTIMRI
1241 E492 SF E7          .WORD SYIRQB     ;VTIMR2
1242 E494 BF E7          .WORD SYIRQB     ;VTIMR4
1243 E496 06 E7          .WORD SYIRQ ;VIMIRQ
1244 E498 00 00 00 00    .WORD 0,0,0,0,0 ; CDTMV1-4
1245 E49C 00 00 00 00
1246 E4A0 00 00
1247 E4A2 AE E7          .WORD SYSVBL     ;VVBLKI
1248 E4A4 05 E9          .WORD XITVBL     ;VVBLKD
1249                      ;
1250                      *=$900C
1251                      ;
1252 900C A9 E6          LDA      #PIRQH   ;SET UP RAM VECTORS FOR LINBUG VERSION
1253 900E 8D F9 FF      STA      $FFF9
1254 9011 A9 F3          LDA      #PIRQL
1255 9013 8D F8 FF      STA      *FFFB
1256 9016 A9 E7          LDA      1#PNMIH
1257 9018 8D FB FF      STA      $FFFB
1258 901E A9 91          LDA      #PNMIL
1259 901D 8D FA FF      STA      $FFFA
1260 9020 60            RTS

```