IMSAI

```
;       CBIOS FOR CP/M 2.2 AND ICOM FRUGAL FLOPPY
;       COPYRIGHT 1977,1978,1979 BY BRUCE R. RATOFF - LAST REV 7/13/80
;
FALSE   EQU     0               ; GROUND RULES
TRUE    EQU     NOT FALSE       ; (THAT MAKES SENSE)
BASE    EQU     $               ; TO FIND OUT IF MAC +R OPTION WAS USED
;
MSIZE   EQU     20              ; SET SYSTEM MEMORY SIZE (IN K) HERE
;
; ******** SET DESIRED CONFIGURATION TRUE IN LIST BELOW ********
; WARNING: EXACTLY ONE OF THE I/O EQUATES BELOW MAY BE TRUE.
;          TO DO OTHERWISE WILL PRODUCE "UNPREDICTABLE" RESULTS
CONFA   EQU     FALSE           ; ALTAIR 88-2SIO
CONFB   EQU     FALSE           ; IMSAI SIO2
CONFC   EQU     FALSE           ; ALTAIR SIO A,B,C (NOT REV 0)
CONFD   EQU     FALSE           ; ALTAIR SIO A,B,C (REV 0)
CONFE   EQU     FALSE           ; PROCESSOR TECHNOLOGY 3P+S
CONFF   EQU     FALSE           ; IMSAI MIO
CONFG   EQU     FALSE           ; ALTAIR 88-4PIO
CONFH   EQU     FALSE           ; CROMEMCO TU-ART
CONFS   EQU     FALSE           ; CONSOLE IS VIA PTCO SOLOS
CONFX   EQU     FALSE           ; CONSOLE IS VIA EXIDY SORCERER MONITOR
CONFZ   EQU     FALSE           ; CONSOLE IS VIA ZAPPLE
RELOC   EQU      TRUE           ; TRUE ONLY IF O.E.M. BUILDING MOVCPM...
;                               ; IF RELOC IS TRUE, ALL I/O EQUATES
;                               ; ABOVE MUST BE FALSE
;
  IF CONFA+CONFB+CONFC+CONFD+CONFE+CONFF+CONFG+CONFH+CONFS+CONFX+CONFZ+RELOC+1
  'ERROR - MORE THAN ONE CONFIGURATION SELECTED'
  ENDIF
;
; SOME DEFAULT VALUES FOR NON-CONSOLE DEVICES
RSTAT   EQU     6               ; READER STATUS PORT
RIAND   EQU     1               ; READER AND MASK
RIXOR   EQU     0               ; READER XOR MASK
RDATA   EQU     7               ; READER DATA PORT
PSTAT   EQU     6               ; PUNCH STATUS PORT
POAND   EQU     80H             ; PUNCH AND MASK
POXOR   EQU     0               ; PUNCH XOR MASK
PDATA   EQU     7               ; PUNCH DATA PORT
LSTAT   EQU     0   01          ; LIST STATUS PORT
LOAND   EQU     80H 01          ; LIST AND MASK
LOXOR   EQU     0   01          ; LIST XOR MASK
LDATA   EQU     1   00          ; LIST DATA PORT
;
; DEFAULT VALUES FOR CONSOLE PORTS
        IF      RELOC
CSTAT   EQU     4               ; CONSOLE STATUS PORT
CIAND   EQU     1               ; CONSOLE INPUT AND MASK
CIXOR   EQU     0               ; CONSOLE INPUT XOR MASK
COAND   EQU     80H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     0               ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     5               ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFA
CSTAT   EQU     10H             ; CONSOLE STATUS PORT
CIAND   EQU     1               ; CONSOLE INPUT AND MASK
```

Should be changed

STAT

```
CIXOR   EQU     1               ; CONSOLE INPUT XOR MASK
COAND   EQU     02H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     02H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     11H             ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFB
CSTAT   EQU     03H             ; CONSOLE STATUS PORT
CIAND   EQU     02H             ; CONSOLE INPUT AND MASK
CIXOR   EQU     02H             ; CONSOLE INPUT XOR MASK
COAND   EQU     01H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     01H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     02H             ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFC
CSTAT   EQU     00H             ; CONSOLE STATUS PORT
CIAND   EQU     01H             ; CONSOLE INPUT AND MASK
CIXOR   EQU     00H             ; CONSOLE INPUT XOR MASK
COAND   EQU     80H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     00H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     01H             ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFD
CSTAT   EQU     00H             ; CONSOLE STATUS PORT
CIAND   EQU     20H             ; CONSOLE INPUT AND MASK
CIXOR   EQU     20H             ; CONSOLE INPUT XOR MASK
COAND   EQU     02H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     02H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     01H             ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFE
CSTAT   EQU     00H             ; CONSOLE STATUS PORT
CIAND   EQU     40H             ; CONSOLE INPUT AND MASK
CIXOR   EQU     40H             ; CONSOLE INPUT XOR MASK
COAND   EQU     80H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     80H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     01H             ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFF
CSTAT   EQU     43H             ; CONSOLE STATUS PORT
CIAND   EQU     02H             ; CONSOLE INPUT AND MASK
CIXOR   EQU     02H             ; CONSOLE INPUT XOR MASK
COAND   EQU     01H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     01H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     42H             ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFG
CSTAT   EQU     10H             ; CONSOLE STATUS PORT FOR INPUT
CIAND   EQU     80H             ; CONSOLE INPUT AND MASK
CIXOR   EQU     80H             ; CONSOLE INPUT XOR MASK
COAND   EQU     80H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     80H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     11H             ; CONSOLE DATA PORT FOR INPUT
```

```
CSTATO  EQU     12H             ; STATUS PORT FOR OUTPUT
CDATAO  EQU     13H             ; DATA PORT FOR OUTPUT
        ENDIF
;
        IF      CONFH
CSTAT   EQU     80H             ; CONSOLE STATUS PORT
CIAND   EQU     40H             ; CONSOLE INPUT AND MASK
CIXOR   EQU     40H             ; CONSOLE INPUT XOR MASK
COAND   EQU     80H             ; CONSOLE OUTPUT AND MASK
COXOR   EQU     80H             ; CONSOLE OUTPUT XOR MASK
CDATA   EQU     81H             ; CONSOLE DATA PORT
        ENDIF
;
        IF      CONFA OR CONFB OR CONFC OR CONFD OR CONFE OR CONFF OR CONFH
CSTATO  EQU     CSTAT           ; ALLOWS OUTPUT PORTS TO DIFFER
CDATAO  EQU     CDATA           ; FROM INPUT PORTS (AS IN CONFIG. G)
        ENDIF
;
        IF      CONFS
AINP    EQU     0C022H          ; SOLOS VARIABLE INPUT ROUTINE
AOUT    EQU     0C01CH          ; SOLOS VARIABLE OUTPUT ROUTINE
        ENDIF
;
        IF      CONFX
AINP    EQU     0E018H          ; EXIDY INPUT TO A FROM KEYBD
AOUT    EQU     0E01BH          ; EXIDY OUTPUT TO SCREEN FROM A
        ENDIF
;
        IF      CONFZ
ZCI     EQU     0F003H          ; ZAPPLE CONSOLE INPUT
ZRI     EQU     0F006H          ; READER INPUT
ZCO     EQU     0F009H          ; CONSOLE OUTPUT
ZPO     EQU     0F00CH          ; PUNCH OUTPUT
ZLO     EQU     0F00FH          ; LIST OUTPUT
ZSTAT   EQU     0F012H          ; CONSOLE STATUS
        ENDIF
;
; ******** END OF USER-CONFIGURABLE EQUATES ********
;
;
        IF      RELOC
SSIZE   EQU     0
MLOC    EQU     BASE
        ENDIF
;
        IF      NOT RELOC
SSIZE   EQU     MSIZE           ; NOMINAL MEMORY SIZE OF CP/M
MLOC    EQU     SSIZE*1024-1C00H; BASE ADDRESS OF CP/M
        ENDIF
;
VERS    EQU     22              ; VERSION NUMBER TIMES TEN
PATCH   EQU     MLOC+1600H      ; STARTING ADDRESS OF BIOS
FACTOR  EQU     3               ; SKEW FACTOR TO USE DURING BOOT OPERATIONS
;                               ;   WARNING: FACTOR MUST BE AN ODD NUMBER
;
DATAI   EQU     0C0H            ; ICOM DATA/STATUS INPUT PORT
DATAO   EQU     0C1H            ; ICOM DATA OUTPUT PORT
CNTRL   EQU     0C0H            ; ICOM COMMAND OUTPUT PORT
```

```
;
          ORG     PATCH-BASE
CPMB      EQU     MLOC              ; BASE ADDRESS OF CP/M (CCP)
BDOS      EQU     MLOC+806H         ; ENTRY ADDRESS OF BDOS
CPML      EQU     $-CPMB            ; LENGTH OF CCP+BDOS
NSECTS    EQU     2*26              ; TOTAL SECTORS IN BOOT TRACKS
BUFF      EQU     80H               ; DEFAULT DMA BUFFER
RTRY      EQU     10                ; NUMBER OF RETRIES TO DO ON A READ
;
COLDBT:   JMP     BOOT              ; COLD BOOT ENTRY
WBOTE:    JMP     WBOT              ; WARM BOOT ENTRY
          JMP     CSTS              ; CONSOLE STATUS TO A REG
CI:       JMP     CONIN             ; CONSOLE INPUT TO A REG
CO:       JMP     COUT              ; CONSOLE OUTPUT FROM C REG
          JMP     LO                ; PRINTER OUTPUT FROM C REG
          JMP     PO                ; PUNCH OUTPUT FROM C REG
          JMP     READER            ; READER INPUT TO A REG
          JMP     HOME              ; REQUEST TRACK 0
          JMP     SELDSK            ; REQUEST DRIVE IN C REG, RETURN DPB IN HL
          JMP     SETTRK            ; REQUEST TRACK IN C REG
          JMP     SETSEC            ; REQUEST SECTOR IN C REG
          JMP     SETDMA            ; REQUEST DMA ADDRESS IN BC PAIR
          JMP     READ              ; READ A SECTOR WITH CURRENT REQUEST VALUES
          JMP     WRITE             ; WRITE A SECTOR  "     "       "     "
          JMP     LSTS              ; PRINTER STATUS TO A REG
          JMP     SECTRAN           ; TRANSLATE SECTOR # FROM BC TO HL
;
; DISK PARAMETER BLOCKS
;
NDISKS    EQU     4                 ; NUMBER OF DRIVES
DPBASE    EQU     $
DPE0:     DW      XLT0,0            ; TRANSLATE TABLE
          DW      0,0
          DW      DIRBUF,DPB0       ; DIRECTORY BUFFER, PARAMETER BLOCK
          DW      CSV0,ALV0         ; CHECK VECTOR, ALLOCATION VECTOR
DPE1:     DW      XLT1,0            ; TRANSLATE TABLE
          DW      0,0
          DW      DIRBUF,DPB1       ; DIRECTORY BUFFER, PARAMETER BLOCK
          DW      CSV1,ALV1         ; CHECK VECTOR, ALLOCATION VECTOR
DPE2:     DW      XLT2,0            ; TRANSLATE TABLE
          DW      0,0
          DW      DIRBUF,DPB2       ; DIRECTORY BUFFER, PARAMETER BLOCK
          DW      CSV2,ALV2         ; CHECK VECTOR, ALLOCATION VECTOR
DPE3:     DW      XLT3,0            ; TRANSLATE TABLE
          DW      0,0
          DW      DIRBUF,DPB3       ; DIRECTORY BUFFER, PARAMETER BLOCK
          DW      CSV3,ALV3         ; CHECK VECTOR, ALLOCATION VECTOR
;
; DISK CONFIGURATION BLOCK
;
DPB0:     DW      26                ; SECTORS PER TRACK
          DB      3                 ; BLOCK SHIFT
          DB      7                 ; BLOCK MASK
          DB      0                 ; EXTENT MASK
          DW      242               ; HIGHEST BLOCK NUMBER
          DW      63                ; HIGHEST DIRECTORY ENTRY NUMBER
          DB      11000000B         ; FIRST ALLOCATION VECTOR
          DB      0                 ; SECOND ALLOCATION VECTOR
```

```
        DW      16              ; SIZE OF CHECK BUFFER
        DW      2               ; NUMBER OF SYSTEM TRACKS
;
XLT0:   DB      1,7,13,19,25,5,11,17    ; SECTOR TRANSLATE TABLE
        DB      23,3,9,15,21,2,8,14
        DB      20,26,6,12,18,24,4,10
        DB      16,22
;
ALS0    EQU     31              ; SIZE OF ALLOCATION VECTOR
CSS0    EQU     16              ; SIZE OF CHECKSUM VECTOR
;
DPB1    EQU     DPB0            ; PARAMS FOR REMAINING DRIVES ARE THE SAME
ALS1    EQU     ALS0            ; LIKEWISE ALLOCATION VECTOR SIZE
CSS1    EQU     CSS0            ; CHECKSUM VECTOR SIZE
XLT1    EQU     XLT0            ; TRANSLATE TABLE
;
DPB2    EQU     DPB0            ; PARAMS FOR REMAINING DRIVES ARE THE SAME
ALS2    EQU     ALS0            ; LIKEWISE ALLOCATION VECTOR SIZE
CSS2    EQU     CSS0            ; CHECKSUM VECTOR SIZE
XLT2    EQU     XLT0            ; TRANSLATE TABLE
;
DPB3    EQU     DPB0            ; PARAMS FOR REMAINING DRIVES ARE THE SAME
ALS3    EQU     ALS0            ; LIKEWISE ALLOCATION VECTOR SIZE
CSS3    EQU     CSS0            ; CHECKSUM VECTOR SIZE
XLT3    EQU     XLT0            ; TRANSLATE TABLE
;
SIGNON: DB      13,10           ; MESSAGE TO PRINT DURING COLD BOOT
        DB      MSIZE/10+'0',MSIZE MOD 10+'0'
        DB      'k CP/M vers '
        DB      VERS/10+'0','.',VERS MOD 10+'0'
        DB      0
;
BOOT:   LXI     SP,BUFF+80H     ; SET A TEMPORARY STACK
        CALL    IOINIT          ; INITIALIZE SIO'S, PIO'S, ETC.
        CALL    INITLD          ; GO LOAD/EXECUTE INITIALIZER PROGRAM
        LXI     H,SIGNON        ; PRINT SIGNON
        CALL    PRMSG
        SUB     A               ; SELECT DRIVE A, USER 0
        STA     4
        LXI     H,CPMB          ; SET UP TO USE CCP COLD ENTRY
        JMP     GOCPM           ; GO SET UP VECTORS AND START CP/M
;
WBOT:   LXI     SP,BUFF         ; COME HERE FOR WARM BOOT
        LDA     DISKN           ; SAVE LOGGED IN USER/DRIVE
        STA     DISKT
WBOT0:  LXI     H,CPMB-128      ; BASE OF CP/M MINUS LENGTH OF BOOT SECTOR
        SHLD    IOD
        LXI     H,100H          ; TRACK 0 SECTOR 1
        SHLD    IOT
        MVI     C,0             ; DRIVE 0
        CALL    SELDSK
        MVI     B,NSECTS        ; TOTAL COUNT OF SECTORS TO LOAD
RDSC:   PUSH    B
        LDA     IOD+1           ; SKIP AREA USED FOR BIOS
        CPI     PATCH/256
        JNC     RDSC1
        CALL    READ            ; READ NEXT SECTOR
        JNZ     WBOT            ; WE'RE HAVING TROUBLE?
```

```
RDSC1:  LHLD    IOD             ; OTHERWISE, BUMP LOAD ADDRESS
        LXI     D,128*FACTOR    ;   BY SECTOR SIZE TIMES SKEW FACTOR
        DAD     D
        LDA     IOS             ; BUMP SECTOR NUMBER BY SKEW FACTOR
        ADI     FACTOR
        CPI     27              ; MODULO 26 (SECTORS PER TRACK)
        JC      NXSEC           ; WRAP NOT NEEDED
        JZ      NXTRK           ; EXACTLY 27 MEANS WE'VE GOT END OF TRACK
        LXI     D,-128*26       ; WRAP LOAD ADDRESS
        DAD     D
NXTRK:  SUI     26              ; WRAP SECTOR NUMBER
NXSEC:  STA     IOS             ; STORE NEW PARAMETERS
        SHLD    IOD
        DCR     A               ; IF SECTOR # NOW = 1, BUMP TRACK
        JNZ     IFMORE          ; ELSE JUST SEE IF MORE SECTORS NEEDED
        LDA     IOT
        MOV     C,A             ; BUMP TRACK NUMBER
        INR     C
        CALL    SETTRK
IFMORE: POP     B               ; STILL MORE SECTORS?
        DCR     B
        JNZ     RDSC            ; IF NOT, GO TO CP/M AT WARM ENTRY POINT
        LXI     H,CPMB+3
;
; THIS ROUTINE SETS UP THE VECTORS AT 0 AND 5 AND THE DEFAULT DMA ADDRESS
; IT THEN BRANCHES TO (HL) WITH THE LAST LOGGED-IN USER/DRIVE IN THE C REG
;
GOCPM:  PUSH    H               ; SAVE BRANCH ADDRESS
        LXI     B,BUFF          ; SET DEFAULT DMA ADDR
        CALL    SETDMA
        MVI     A,0C3H          ; STORE JMP OPCODE AT 0
        STA     0
        LXI     H,WBOTE         ; WARM BOOT ADDRESS AT 1&2
        SHLD    1
        STA     5               ; JMP OPCODE AT 5
        LXI     H,BDOS          ; AND BDOS ENTRY AT 6&7
        SHLD    6
        LDA     4               ; PICK UP USER/DRIVE SAVED BY CCP
        MOV     C,A             ; PASS IN C REG
        RET                     ; GO TO ADDR SAVED AT GOCPM
;
; THIS ROUTINE IS CALLED TO SELECT A DRIVE.  IT FIRST CHECKS THAT THE NUMBER
; OF DEFINED DRIVES HAS NOT BEEN EXCEEDED, RETURNING WITH HL SET TO 0 TO
; INDICATE AN ERROR.  IF VALID, THE DRIVE NUMBER IS STORED AT DISKN AND THE
; ADDRESS OF THE DISK PARAMETER BLOCK FOR THAT DRIVE IS RETURNED IN HL.
;
SELDSK: LXI     H,0             ; SET UP TO RETURN ERROR
        MOV     A,C
        CPI     NDISKS          ; CHECK FOR MAX DISK NO.
        RNC
        STA     DISKN           ; IT'S VALID...STORE IT AT DISKN
        MVI     A,0FFH
        STA     SEEKFLG         ; SET SEEKFLG
        MOV     L,C             ; CALCULATE ADDRESS OF DISK PARAMETER BLOCK
        DAD     H
        DAD     H
        DAD     H
        DAD     H
```

```
        LXI     D,DPBASE
        DAD     D
        RET
;
HOME:   MVI     C,0             ; REQUEST TRACK 0
;
; REQUEST TRACK GIVEN IN C REG
SETTRK: LXI     H,SEEKFLG       ; SET SEEK FLAG
        MVI     M,0FFH
        INX     H               ; STORE TRACK NUMBER
        MOV     M,C
        RET
;
SETSEC: LXI     H,IOS           ; SET SECTOR NUMBER FROM C REG
        MOV     M,C
        RET
;
SETDMA: MOV     H,B             ; SET DMA ADR FROM BC
        MOV     L,C
        SHLD    IOD
        RET
;
SECTRAN:XCHG                    ; TRANSLATE LOGICAL SECTOR IN BC USING TABLE
        DAD     B               ;   AT (HL).  RETURN RESULT IN HL.
        MOV     L,M
        RET
;
; THIS IS THE PRIMARY ROUTINE TO READ A SECTOR USING THE CURRENT PARAMETERS.
; IT WILL TRY UP TO RTRY TIMES, DOING A HOME AND A RE-SEEK EVERY 4 TIMES.
; IF SECTOR IS UNREADABLE, IT RETURNS WITH A=1.  IF SUCCESSFUL, A=0.
;
READ:   MVI     C,RTRY          ; SET UP RETRY COUNTER
READ1:  CALL    STUP            ; CALL ROUTINE TO SET UP DRIVE/TRACK/SECTOR
        MVI     A,3             ; SEND READ COMMAND TO CONTROLLER
        CALL    LOOP
        IN      DATAI           ; CHECK STATUS FOR CRC ERROR
        ANI     8
        JZ      RDOK            ; BRANCH IF NO ERROR
        DCR     C               ; ELSE COUNT DOWN RETRY COUNTER
        MVI     A,1             ; QUIT IF EXHAUSTED
        RZ
        MOV     A,C             ; IS IT TIME FOR A RE-SEEK?
        ANI     3               ;  (INDICATED BY LOW BITS OF C = 3)
        JPO     READ1
        MVI     A,0FFH          ; DOING RE-SEEK...SET SEEK FLAG
        STA     SEEKFLG         ;   SO THAT NEXT CALL TO STUP WILL SEEK
        CALL    RESET           ; HOME THE CURRENT DRIVE
        JMP     READ1           ; AND GO RE-TRY THE READ
RDOK:   MVI     C,128           ; GOT THE DATA...SET COUNTER TO SECTOR SIZE
        LHLD    IOD             ; POINT TO CURRENT DMA ADDRESS
RDLUP:  MVI     A,40H           ; REQUEST A BYTE FROM CONTROLLER BUFFER
        OUT     CNTRL
        IN      DATAI           ; GET THE BYTE
        MOV     M,A             ; STORE IT
        INX     H               ; BUMP POINTER
        MVI     A,41H           ; SHIFT CONTROLLER BUFFER TO NEXT BYTE
        CALL    LOOP
        DCR     C               ; LOOP UNTIL 128 BYTES TRANSFERRED
```

```
        JNZ     RDLUP
        SUB     A               ; SAY NO ERRORS AND EXIT
        RET
;
; THIS IS THE PRIMARY ROUTINE TO WRITE A SECTOR USING THE CURRENT PARAMETERS.
; BEFORE PERFORMING THE WRITE IT WILL CHECK THAT THE CURRENT DISK IS NOT
; WRITE-PROTECTED, AND IF IT IS, WILL GIVE THE USER A CHANCE TO CORRECT
; OR ABORT.  IT ALWAYS RETURNS A=0, INDICATING NO ERRORS.
;
WRITE:  MVI     C,128           ; SET COUNTER TO SECTOR SIZE
        LHLD    IOD             ; POINT TO CURRENT DMA ADDRESS
WRLUP:  MOV     A,M             ; FETCH A BYTE
        OUT     DATAO           ; SHIFT IT INTO THE CONTROLLER'S BUFFER
        MVI     A,31H           ; ISSUE SHIFT BUFFER COMMAND
        CALL    LOOP
        INX     H               ; BUMP POINTER
        DCR     C               ; LOOP UNTIL WHOLE SECTOR MOVED TO BUFFER
        JNZ     WRLUP
RTRYP:  CALL    STUP            ; SET UP UNIT/TRACK/SECTOR
        IN      DATAI           ; CHECK FOR WRITE PROTECT
        ANI     10H
        JZ      TRYWR           ; BRANCH IF OK TO PROCEED
        LXI     H,PRTMSG        ; OTHERWISE, PRINT MESSAGE AND GET RESPONSE
        CALL    ERROR
        JMP     RTRYP           ; IF WE GOT BACK FROM ERROR, RETRY WRITE
TRYWR:  MVI     A,5             ; SEND WRITE COMMAND TO CONTROLLER
        CALL    LOOP
WROK:   SUB     A               ; CLEAR ERROR FLAG AND EXIT
        RET
;
; THIS IS A GENERAL ROUTINE TO PRINT A MESSAGE FROM (HL) UNTIL M(HL)=0
;
PRMSG:  MOV     A,M             ; FETCH A BYTE
        ORA     A               ; CHECK FOR ZERO
        RZ                      ; RETURN IF ZERO
        PUSH    H               ; ELSE SAVE POINTER
        MOV     C,A             ; PRINT THE BYTE
        CALL    CO
        POP     H               ; RESTORE POINTER
        INX     H               ; BUMP IT
        JMP     PRMSG           ; LOOP BACK FOR MORE
;
; THIS ROUTINE IS CALLED WHEN AN ERROR HAS OCCURRED WHICH MAY BE FIXABLE
; BY THE OPERATOR.  IT WILL DISPLAY A MESSAGE FROM (HL) AND THEN WAIT FOR
; ONE CHARACTER FROM THE CONSOLE.  IF THE CHARACTER IS CONTROL-C, THE
; LOGGED IN USER/DRIVE IS FORCED TO ZERO AND A WARM BOOT IS PERFORMED.
; ALL OTHER RESPONSES CAUSE A RETURN TO THE CALLER, WHO WILL USUALLY RETRY THE
; OPERATION WHICH CAUSED THE ERROR.
;
ERROR:  CALL    PRMSG           ; DISPLAY THE MESSAGE
        CALL    CI              ; GET A RESPONSE
        MOV     B,A
        MVI     C,13            ; ECHO CRLF
        CALL    CO
        MVI     C,10
        CALL    CO
        MVI     A,3             ; CHECK FOR CONTROL-C
        SUB     B
```

```
        RNZ                     ; IF NOT CONTROL-C, RETURN
        STA     4               ; ELSE KILL USER/DRIVE AND REBOOT
        JMP     WBOT
;
; THIS IS A GENERALIZED ROUTINE WHICH IS USED TO SET UP FOR BOTH READS AND
; WRITES.  IT SELECTS THE UNIT, TRACK AND SECTOR GIVEN BY DISKN, IOT AND IOS.
; IF THE SEEK FLAG IS SET, A SEEK IS PERFORMED.  OTHERWISE IT IS NOT.  NOTE
; ALSO THAT A SEEK TO TRACK ZERO IS CONVERTED TO A HOME COMMAND.
;
STUP:   MVI     A,0BH           ; SEND CLEAR ERROR FLAGS COMMAND
        CALL    LOOP
        LXI     H,IOS           ; POINT TO SECTOR NUMBER
        LDA     DISKN           ; FETCH DRIVE NUMBER
        RRC
        RRC
        ANI     0C0H            ; MOVE INTO CORRECT BIT POSITIONS
        ORA     M               ; COMBINE UNIT/SECTOR FOR CONTROLLER
        OUT     DATA0
        MVI     A,21H           ; ISSUE SET UNIT/SECTOR COMMAND
        CALL    LOOP
        MVI     B,100           ; TENTHS OF A SECOND TO WAIT IF NOT READY
STUP0:  LXI     D,8000          ; DELAY COUNT FOR ONE TENTH OF A SECOND
STUP1:  IN      DATAI           ; GET DRIVE STATUS
        ANI     20H             ; TEST 'DRIVE FAIL' BIT (INDICATES READY)
        JZ      STUP2           ; BRANCH IF READY
        DCX     D               ; COUNT DOWN SHORT DELAY
        MOV     A,D             ; TEST FOR ZERO
        ORA     E
        JNZ     STUP1           ; NOT YET
        DCR     B               ; COUNT DOWN LONG DELAY
        JNZ     STUP0
        LXI     H,RDYMSG        ; DELAY EXHAUSTED...COMPLAIN
        CALL    ERROR
        JMP     STUP            ; RETRY THE WHOLE SETUP
STUP2:  DCX     H               ; DRIVE IS READY...POINT TO SEEK FLAG
        DCX     H
        INR     M               ; TEST FOR 0FFH
        MVI     M,0             ; ALWAYS CLEAR IT ANYWAY
        RNZ                     ; DONE IF FLAG WASN'T SET
        INX     H               ; FLAG WAS SET...POINT TO IOT
        MOV     A,M             ; GET TRACK NUMBER
        ANA     A               ; TEST FOR ZERO
        JZ      RESET           ; BRANCH IF HOME INSTEAD OF SEEK
        OUT     DATA0           ; SEND TRACK NUMBER TO CONTROLLER
        MVI     A,11H           ; ISSUE LOAD TRACK REGISTER COMMAND
        CALL    LOOP
        MVI     A,9             ; ISSUE SEEK COMMAND
                                ; FALL THRU TO 'LOOP'
;
; THIS IS A GENERAL ROUTINE TO OUTPUT A COMMAND FROM THE A REG AND WAIT
; UNTIL THE BUSY FLAG CLEARS.
;
LOOP:
        OUT     CNTRL           ; ISSUE COMMAND
        SUB     A               ; FOLLOWED BY 0
        OUT     CNTRL
LOOP1:  IN      DATAI           ; CHECK FOR BUSY
        RAR
```

```
        JC      LOOP1           ; LOOP TILL NOT BUSY
        RET
;
RESET:  MVI     A,81H           ; RESET THE CONTROLLER
        CALL    LOOP
        MVI     A,0DH           ; AND HOME THE CURRENT DRIVE
        JMP     LOOP
;
RDYMSG: DB      13,10,'NOT READY',0
PRTMSG: DB      13,10,'PROTECTED',0
;
; THIS ROUTINE SCANS THE CURRENT DRIVE FOR AN INITIALIZATION PROGRAM,
; AND LOADS AND EXECUTES IT IF IT IS FOUND.
; IT THEN DISABLES ITSELF TO PREVENT BEING CALLED AGAIN, IN CASE OTHER
; BRANCHES TO THE COLD BOOT VECTOR OCCUR.
; THIS CODE MAY BE REMOVED OR AMENDED AS REQUIRED BY THE SYSTEM.
; THE NAME OF THE INITIALIZE PROGRAM MAY BE CHANGED BY ALTERING THE
; CHARACTER STRING AT 'FCB'.  NOTE THAT 'FCB' TAKES THE FORM OF A
; STANDARD CP/M SEQUENTIAL FILE CONTROL BLOCK.
;
INITLD: MVI     A,0C9H          ; REPLACE 'MVI A' WITH 'RET'
        STA     INITLD          ; SO WE CAN'T BE EXECUTED TWICE
        MVI     C,0DH           ; RESET THE BDOS
        CALL    BDOS
        LXI     D,FCB
        MVI     C,0FH           ; TRY TO OPEN INITIALIZER FILE
        CALL    BDOS
        INR     A               ; CHECK FOR ERROR (A=0FFH)
        RZ                      ; SKIP LOAD ROUTINE IF FILE NOT FOUND
        SUB     A               ; CLEAR RECORD POINTER IN FCB
        STA     FCB+32
        LXI     D,100H          ; POINT TO START OF TRANSIENT PROGRAM AREA
LDLUP:  PUSH    D
        MVI     C,1AH           ; SET DMA ADDRESS FOR NEXT READ
        CALL    BDOS
        LXI     D,FCB           ; TRY TO READ
        MVI     C,14H
        CALL    BDOS
        POP     D               ; BUMP DMA POINTER
        LXI     H,128
        DAD     D
        XCHG
        ORA     A               ; DID READ GIVE AN ERROR
        JZ      LDLUP           ; LOOP BACK IF NOT
        LXI     H,100H          ; TELL GOCPM TO JUMP TO 100 AFTER SETTING
        JMP     GOCPM           ; THE BIOS AND BDOS VECTORS
;
; THE REMAINING SPACE BETWEEN HERE AND PATCH+380H MAY BE USED BY THE USER
; TO INSERT ANY SPECIAL I/O DRIVERS HE MAY REQUIRE.  THE LABELS CSTS, CONIN,
; COUT, LO, PO, READER AND LSTS ARE POINTED TO DIRECTLY FROM THE JUMP TABLE
; AT THE BEGINNING OF THE BIOS.  ALL OTHER REFERENCES TO THESE ROUTINES PASS
; THROUGH THE JUMP TABLE.
;
PORTIO EQU CONFA OR CONF8 OR CONFC OR CONFD OR CONFE OR CONFF OR CONF0 OR CONFH
;
; CONSOLE STATUS MUST RETURN A=FF IF A CHARACTER IS AVAILABLE, ELSE A=0
CSTS:
        IF      RELOC
```

```
        LXI     H,CCHAR
        IN      CSTAT
        LXI     H,ST1
        ANI     CIAND
        XRI     CIXOR
        MVI     A,0
        RNZ
        CMA
ST1:    RET
        DS      2
        ENDIF
;
        IF      PORTIO
        IN      CSTAT
        ANI     CIAND
        XRI     CIXOR
        MVI     A,0
        RNZ
        CMA
        RET
        ENDIF
;
        IF      CONFS OR CONFX
        LXI     H,CCHAR         ; POINT TO SAVED CHARACTER (IF ANY)
        MOV     A,M             ; SEE IF THERE WAS SOMETHING SAVED
        ANA     A
        JNZ     ST1
        LDA     3               ; NOTHING SAVED...CALL SOLOS
        CALL    AINP
        RZ                      ; SOLOS ALSO HAS NOTHING...RETURN FALSE
        MOV     M,A             ; SOLOS RETURNED A CHARACTER...SAVE IT
ST1:    MVI     A,0FFH          ; RETURN TRUE
        RET
        ENDIF
;
        IF      CONFZ
        JMP     ZSTAT
        ENDIF
;
; CONSOLE INPUT SHOULD FETCH A KEYBOARD CHARACTER, STRIP PARITY, AND RETURN
; THE CHARACTER IN THE A REG.
CONIN:
        IF      RELOC
        CALL    CSTS
        ORA     A
        JZ      CONIN
        MOV     A,M
        IN      CDATA
        ANI     7FH
        RET
        ENDIF
;
        IF      PORTIO
        CALL    CSTS
        ORA     A
        JZ      CONIN
        IN      CDATA
        ANI     7FH
```

```
        RET
        ENDIF
;
        IF      CONFS OR CONFX
        CALL    CSTS
        ORA     A
        JZ      CONIN
        MOV     A,M
        MVI     M,0
        RET
        ENDIF
;
        IF      CONFZ
        JMP     ZCI
        ENDIF
;
; CONSOLE OUTPUT SHOULD DISPLAY THE CHARACTER IN THE C REG
COUT:
        IF      RELOC
        LDA     LCHAR
        CMP     C
        JNZ     CO1
        CPI     0DH
        NOP
CO1:
        IN      CSTAT
        ANI     COAND
        XRI     COXOR
        NOP ! NOP
        STA     LCHAR
        NOP
        JNZ     COUT
        MOV     A,C
        OUT     CDATA
        RET
        ENDIF
;
        IF      PORTIO
        IN      CSTATO
        ANI     COAND
        XRI     COXOR
        JNZ     COUT
        MOV     A,C
        OUT     CDATAO
        RET
        ENDIF
;
        IF      CONFS
        LDA     LCHAR
        CMP     C
        JNZ     CO1
        CPI     0DH
        RZ
CO1:    PUSH    B
        MOV     B,C
        LDA     3
        CALL    AOUT
        LXI     H,LCHAR
```

```
        POP     B
        MOV     M,C
        RET
        ENDIF
;
        IF      CONFX
        MOV     A,C
        JMP     AOUT
        ENDIF
;
        IF      CONFZ
        JMP     ZCO
        ENDIF
;
; LIST OUTPUT SHOULD SEND THE CHARACTER IN THE C REG TO THE PRINTER
LO:
        IF      RELOC OR CONFS
        LDA     3
        ENDIF
LO1:
        IF      RELOC OR PORTIO
        IN      LSTAT
        ANI     LOAND
        XRI     LOXOR
        JNZ     LO
        MOV     A,C
        OUT     LDATA
        RET
        ENDIF
;
        IF      CONFS
        RLC
        RLC
        MOV     B,C
        JMP     AOUT
        ENDIF
;
        IF      CONFZ
        JMP     ZLO
        ENDIF
;
; PUNCH OUTPUT SENDS THE C REG TO THE PUNCH DEVICE
PO:
        IF      RELOC OR PORTIO
        IN      PSTAT
        ANI     POAND
        XRI     POXOR
        ENDIF
        IF      RELOC
        LXI     H,LO1
        ENDIF
        IF      RELOC OR PORTIO
        JNZ     PO
        MOV     A,C
        OUT     PDATA
        RET
        ENDIF
;
```

```
        IF      CONFS
        LDA     3
        RLC
        RLC
        JMP     LO1
        ENDIF
;
        IF      CONFZ
        JMP     ZPO
        ENDIF
;
; READER INPUT MUST FETCH A CHARACTER FROM THE READER DEVICE TO THE A REG.
; OR RETURN CONTROL-Z (1AH) TO INDICATE END OF FILE.
READER:
        IF      RELOC OR PORTIO
        IN      RSTAT
        ANI     RIAND
        XRI     RIXOR
        ENDIF
        IF      RELOC
        NOP ! NOP
        ENDIF
        IF      RELOC OR PORTIO
        JNZ     READER
        IN      RDATA
        RET
        ENDIF
;
        IF      CONFS
        LDA     3
        RRC
        RRC
        CALL    AINP
        JZ      READER
        RET
        ENDIF
;
        IF      CONFZ
        CALL    ZRI
        RNC
        MVI     A,1AH
        RET
        ENDIF
;
; LIST STATUS MUST RETURN A=FF IF THE PRINTER IS READY FOR A CHARACTER,
; OR A=0 IF THE PRINTER IS BUSY.
LSTS:
        MVI     A,0FFH
        RET
        DS      10
;
; IOINIT MUST PERFORM ANY REQUIRED DEVICE INITIALIZATION SUCH AS SETTING
; BAUD RATES OR PROGRAMMING USART'S OR PIO'S
IOINIT:
        IF CONFC OR CONFD OR CONFE OR CONFX OR CONFZ
        RET
        ENDIF
;
```

```
        IF      RELOC OR CONFS
        SUB     A
        STA     CCHAR
        STA     LCHAR
        STA     3
        RET
        ENDIF
;
        IF      RELOC
        DS      6
        ENDIF
;
        IF      CONFA
        MVI     A,3
        OUT     CSTAT
        MVI     A,11H
        OUT     CSTAT
        RET
        ENDIF
;
        IF      CONFB
        MVI     A,0AAH
        OUT     CSTAT
        MVI     A,40H
        OUT     CSTAT
        MVI     A,0CEH
        OUT     CSTAT
        MVI     A,27H
        OUT     CSTAT
        RET
        ENDIF
;
        IF      CONFF
        SUB     A
        OUT     CSTAT
        RET
        ENDIF
;
        IF      CONFG
        SUB     A
        OUT     CSTAT
        OUT     CDATA
        OUT     CSTAT0
        CMA
        OUT     CDATA0
        MVI     A,24H
        OUT     CSTAT
        OUT     CSTAT0
        RET
        ENDIF
;
        IF      CONFH
        MVI     A,1             ; RESET 5501
        OUT     CSTAT+2
        SUB     A               ; DISABLE 5501 INTERRUPTS
        OUT     CSTAT+3
        MVI     A,40H           ; SELECT 9600 BAUD
        OUT     CSTAT
```

```
        RET
        ENDIF
;
FIRST$FREE  EQU     $
;
; THE SPACE FROM HERE TO 'FCB' IS AVAILABLE FOR USER PATCHES.
;
        DS      COLDBT+370H-$           ; DECLARE FREE AREA DYNAMICALLY

                                        ; THE ADDRESS 'FCB' IS THE LAST 14
                                        ; BYTES BROUGHT IN BY THE COLD BOOT.
FCB:    DB      0,'INITIAL COM',0,0,0,0 ; THIS IS THE FILE CONTROL BLOCK FOR
        DS      17                      ; THE INITIALIZE PROGRAM.  IT MUST
                                        ; REMAIN AT THIS ADDRESS TO MAKE THE
                                        ; BEST USE OF AVAILABLE BIOS SPACE
;
; THE FOLLOWING EIGHT VARIABLES CONSTITUTE THE WORKING STORAGE FOR THE BIOS
;
DISKT   DS      1               ; HOLDS LOGGED IN USER/DRIVE DURING WARM BOOT
DISKN   DS      1               ; CURRENT DRIVE
SEEKFLG DS      1               ; SET TO OFFH WHEN A SEEK IS NEEDED
IOT     DS      1               ; CURRENT TRACK
IOS     DS      1               ; CURRENT SECTOR
IOD     DS      2               ; CURRENT DISK I/O ADDRESS
CCHAR   DS      1               ; TEMP FOR SOLOS CONSOLE STATUS & INPUT
LCHAR   DS      1               ; TEMP FOR SOLOS CONSOLE OUTPUT
;
; THE AREA FROM HERE TO THE NOMINAL END OF RAM IS USED BY THE BIOS FOR
; SCRATCH AND TABLE STORAGE.
;
BEGDAT  EQU     $
;
DIRBUF: DS      128             ; DIRECTORY ACCESS BUFFER
ALV0:   DS      ALS0            ; ALLOCATION VECTOR DRIVE 0
CSV0:   DS      CSS0            ; CHECKSUM VECTOR DRIVE 0
ALV1:   DS      ALS1            ; ALLOCATION VECTOR DRIVE 1
CSV1:   DS      CSS1            ; CHECKSUM VECTOR DRIVE 1
ALV2:   DS      ALS2            ; ALLOCATION VECTOR DRIVE 2
CSV2:   DS      CSS2            ; CHECKSUM VECTOR DRIVE 2
ALV3:   DS      ALS3            ; ALLOCATION VECTOR DRIVE 3
CSV3:   DS      CSS3            ; CHECKSUM VECTOR DRIVE 3
ENDDAT  EQU     $
DATSIZ  EQU     $-BEGDAT
;
        END
```