

``` *TITLE XMON/D VER 1.6 JUMP TABLE FOR MONITOR FUNCTIONS ;***** ```

```

;
; XMON/D - DISKETTE BASED MONITOR
; SUPPORTING MEPLA
;

```

```

; DATE: 24/07/1979 M.J.Herd
;

```

```

; THIS REVISION LEVEL 6
;

```

``` ;***** ; ; SYSTEM ERROR CODES ; ```

```

sec38 equ 38h
;

```

``` ; PUBLIC DECLARATIONS ; ```

```

public bridge
public tti,tto,ttonc,cursor,mon
public error,print,dtst,dcal,dsrd
public dswr,dSDL,dsrda,dswra,indisp
public dswt,dcalw,dsrdw,dswrw,dSDLw
public dsraw,dswaw,rcvi,loadir,load
public ue,link,xmio,hxload,convd
public dblu,gc,crlf,bytut,hxoud,hxinh
public ttyhx,convr,hxine,string,init
PUBLIC JTAB2,JTAB3,JTAB4

```

``` ; ; EXTERNAL REFERENCES ; ```

```

extrn ytti,ytto,yttonc,ycursor,ymon
extrn yerror,yprint,ydtst,ydcal,ydsrd
extrn ydswr,ydSDL,ydsrda,ydswra,yindis
extrn ydswt,ydcalw,ydsrdw,ydswrw,ydSDLw
extrn ydsraw,ydswaw,yrcvi,yloadir,yload
extrn ylink,altjrt,yue,INT6
extrn yxmio,yhxload,convrt
extrn ydblu,ygc,ycrlf,ybytut,yhxoud,yhxinh
extrn yttyhx,yconvr,yhxine,strings,init1

```

```

*E
;

```

```

ORG 40H
;

```

``` ; FIRST JUMP TABLE ; ```

```

JTAB1:
;

```

```

TTI: JMP JTAB2+9
TTO: JMP JTAB2+12
TTONC: JMP JTAB2+15
CURSOR: JMP JTAB2+21
MON: JMP JTAB2+3
ERROR: JMP JTAB2+6
PRINT: JMP JTAB2+24
DTST: JMP JTAB2+51
DCAL: JMP JTAB2+42
DSRD: JMP YDSRD ;Read sector
;A=unit B=trk C=sector HL=buffer
;
DSWR: JMP YDSWR ;Write sector
;A=unit B=trk C=sector HL=buffer
DSDL: JMP YDSDL ;Delete sector
;A=unit B=trk C=sector HL=buffer
DSRDA: JMP YDSRDA ;Read sector and convert
;A=unit B=trk C=sector HL=buffer
DSWRA: JMP YDSWRA ;Write sector and convert
;A=unit B=trk C=sector HL=buffer
INDISP: JMP JTAB2+18
DSWT: JMP JTAB2+48

```

```

;*** The following six routines have
;*** parameters as the corresponding

```

```

;*** ones above and return status in A
DCALW:  JMP      YDCALW      ;Calibrate and wait
DSRDW:  JMP      YDSRDW      ;Read and wait
DSWRW:  JMP      YDSWRW      ;Write and wait
DSDLW:  JMP      YDSDLW      ;Delete and wait
;
DSRAW:  JMP      YDSRAW      ;Read, convert and wait
DSWAW:  JMP      YDSWAW      ;Write converted and wait
RCVI:   JMP      JTAB2+27
LOCDir: JMP      JTAB2+36
LOAD:   JMP      JTAB2+39
LINK:   JMP      YLINK       ;Load program and enter
;
JTABC:   ;Cartridge routine jump table
;Not supported in FROM, but routines may
;be loaded into RAM
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
;
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
CALL     BRIDGE
;
XMIQ:   JMP      JTAB2+30
CALL     BRIDGE
CALL     BRIDGE
HXLOAD: JMP      JTAB2+33
DOBLU:  JMP      YDOBLU      ;Double byte hex output, address in HL
GC:     JMP      YGC         ;Read char from CI, convert to cap, echo
CRLF:   JMP      YCRLF       ;Newline
BYTUT:  JMP      YBYTUT      ;Output A as two hex characters
HXOUD:  JMP      YHXOUD      ;Output DE as four hex chars
HXINH:  JMP      YHXINH      ;Input hex to HL with echo
TTYHX:  JMP      YTTYHX      ;Input hex to A
CONVR:  JMP      YCONVR      ;Convert to hex value
HXINE:  JMP      YHXINE      ;Input hex value to E
UE:     JMP      JTAB2+45
STRING: JMP      STRNG       ;Output character string HL=start, EDT=OFFH
INIT:   JMP      JTAB2
CONVD:  JMP      CONVRT      ;ASCII - EBCDIC conversion routine
;HL = Conversion table start address
;E
;
ORG      100H
;
;JUMP TABLE FOR MONITOR FUNCTIONS
;
;If an alternative jump table is established, with a start
;address JTAB4, then the location JTAB4-3 must contain
;the address of a routine to be called on system load. A RET
;instruction at this point will re-load the system normally.
;
JTAB3:
JMP      INIT1      ;Re-initialise monitor (RESET and POWER-UP)
JMP      YMON       ;Enter monitor command mode (No return)
JMP      YERROR      ;Display error message:- A=error code
JMP      YTTI        ;Read one keyboard char:- A=byte
JMP      YTT0        ;Write one character to screen:- A=byte

```

```

JMP      YTT0      ;Write one character to screen:- A=byte
JMP      YTT0NC    ;Write one character, no control
JMP      YINDISP    ;Read character from display:- A=byte
JMP      YCURSOR    ;Position cursor:- H=line L=char
JMP      YPRINT     ;Print character
JMP      ████████   ;Read from host
JMP      ████████   ;Send to host
JMP      YHXLOAD    ;Load hex program (XMON/S and /SD only)
JMP      YLOCDIR    ;Locate directory entry on an
                    ; Intel-formatted diskette:-
                    ; A=UNIT HL=address of name
                    ; (9 characters, dot removed)
JMP      YLOAD      ;Load program
JMP      YDCAL      ;Calibrate diskette:- A=unit
JMP      YUE        ;Common diskette routine
JMP      YDSWT      ;Wait for completion:- A=status
JMP      YDTST      ;Test if diskette finished

```

⌘E

;
CSEG

HASHING TABLE FOR REROUTING ROUTINE

```

;*****
;***** LOCATION FIXED AT 0140H BY ABSOLUTE ADDRESS IN INTO MODULE *****
;***** ASSEMBLER FAULT PREVENTS RELOCATEABLE REFERENCE. *****
;***** INTO MODULE MUST BE CHANGED IF LOCATION MOVED BY RE-LINKING *****
;*****

```

JTAB2:

```

CALL     MONERZ    ;INIT
CALL     MONERZ    ;MON
CALL     MONERZ    ;ERROR
CALL     MONERZ    ;TTI
CALL     MONERZ    ;TTO
CALL     MONERZ    ;TT0NC
CALL     MONERZ    ;INDISP
CALL     MONERZ    ;CURSOR
CALL     MONERZ    ;PRINT
CALL     MONERZ    ;RCVI
CALL     MONERZ    ;XMIO
CALL     MONERZ    ;HXLOAD
CALL     MONERZ    ;LOCDIR
CALL     MONERZ    ;LOAD
CALL     MONERZ    ;DCAL
CALL     MONERZ    ;UE
CALL     MONERZ    ;DSWT
CALL     MONERZ    ;DTST

```

⌘E

REROUTING ROUTINE VIA JUMP TABLE SPECIFIED IN ALTJRT

MONERZ:

```

XTHL     ; PUSH (H), JTAB+3 -> (H)
PUSH     PSW    ;Save flags
PUSH     D
XCHG     ;Save call ident. (JTAB+3) -> (D)
LHLD     ALTJRT ;Fetch jump table offset (JTAB3-JTAB2-3)
BRIDG1:  DAD     D    ;Convert address to current jump table
                    ;(Normally JTAB3)
POP      D
POP      PSW    ;Restore flags
XTHL
RET      ;Enter subroutine via jump table

```

```

;
; CARTRIDGE LINKING ROUTINE
; VIA JUMP TABLE SPECIFIED IN INT6
; Table structure :-
;     INT6 = CTAB
;     CTAB = INTERRUPT ROUTINE ADDRESS
;     CTAB+3 = access from 008EH (JTAB0)

```

```

;
; CTAB+3 = access from 008EH (JTAB)
;
BRIDGE:
    XTHL                ; PUSH (H), JTABC+3 -> (H)
    PUSH PSW            ; Save flags
    PUSH D
    PUSH B
    XCHG                ; Save call ident. JTABC+3 -> (D)
    LHLD INT6           ; Fetch jump table start (CTAB)
    MOV A,H             ; Cartridge routines present ?
    ORA L
    JZ CERR             ; No
    LXI B,-008EH
    DAD B               ; HL = CTAB-JTABC
    POP B
    JMP BRIDGE1         ; Convert address to current jump table

CERR:
    POP B
    POP D
    POP PSW
    POP H               ; Discard link
    MVI A,sec38
    STC
    RET                 ; Return to subroutine call

;
END

```