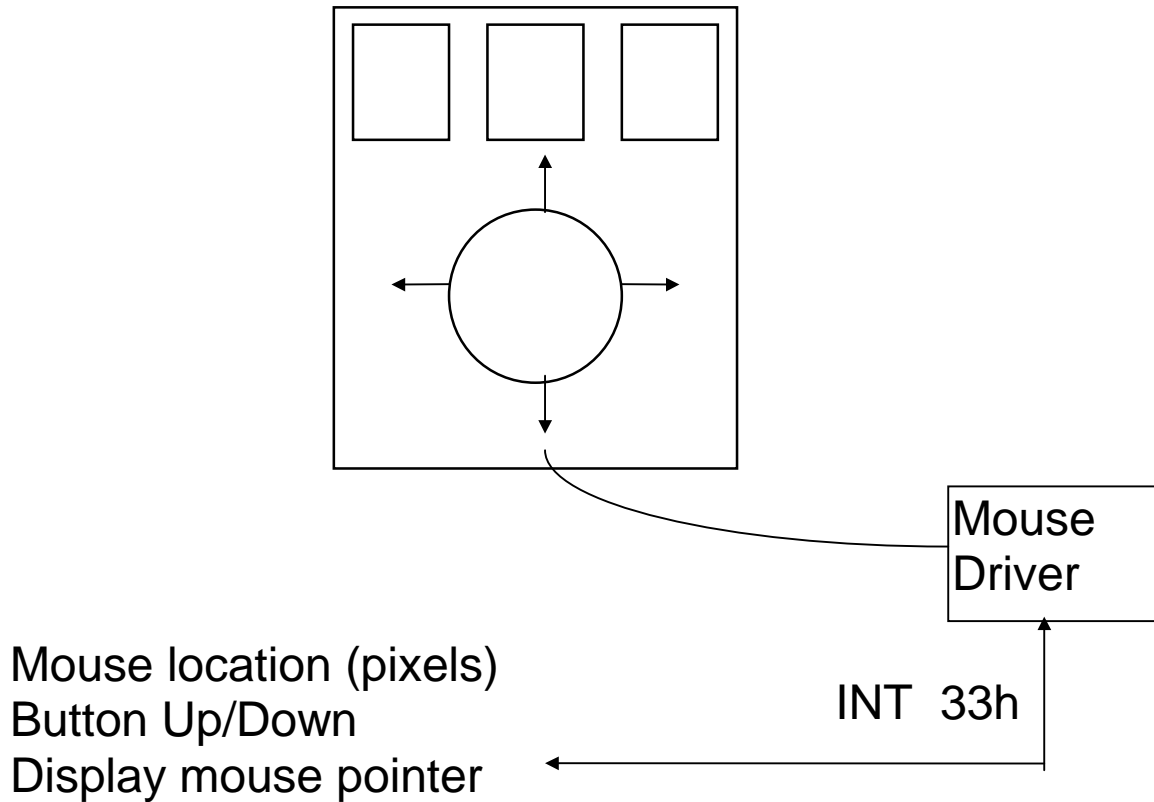


## Lecture 23

### Input/Output: The Mouse and Sound

Text: Chapter 21

## The MOUSE



### INT 33h

- 00h Initialize the Mouse
- 01h Display Mouse Pointer
- 02h Conceal Mouse Pointer
- 03h Get Button Status and Pointer Location
- 04h Set Pointer Location
- 05h Get Button Press Information
- 06h Get Button Release Information
- 07h Set Horizontal Limits for Pointer
- 08h Set Vertical Limits for Pointer

```

TITLE    P21MOUSE (EXE)  Handling the Mouse
        .MODEL SMALL
        .STACK 64
        .DATA
XBINARY DW      0          ;Binary X coordinate
YBINARY DW      0          ;Binary Y coordinate
ASCVAL  DW      ?          ;ASCII field

;        Screen display fields:
DISPDATA LABEL BYTE
XMSG    DB      'X = '     ;X message
XASCII  DW      ?          ;X ASCII value
        DB      ' '       ;
YMSG    DB      'Y = '     ;Y message
YASCII  DW      ?          ;Y ASCII value

        .CODE
BEGIN   PROC FAR
        MOV     AX,@data   ;Initialize
        MOV     DS,AX     ; DS register
        CALL    Q10CLEAR   ;Clear screen
        CALL    B10INIT   ;Initialize mouse
        CMP     AX,00      ;Mouse installed?
        JE      A90       ; no -- exit
A10:
        CALL    D10PTR     ;Get mouse pointer
        CMP     BX,01     ;Button pressed?
        JE      A80       ; yes -- exit
        CALL    Q20CURS   ;Set cursor
        MOV     AX,XBINARY ;
        CALL    G10CONV   ;X to ASCII
        MOV     AX,ASCVAL ;
        MOV     XASCII,AX ;
        MOV     AX,YBINARY ;
        CALL    G10CONV   ;Y to ASCII
        MOV     AX,ASCVAL ;
        MOV     YASCII,AX ;
        CALL    Q30DISP   ;Display X and Y values

```

```

        JMP      A10          ;Repeat
A80:    CALL     H10HIDE      ;Hide mouse pointer
A90:    CALL     Q10CLEAR    ;Clear screen
        MOV     AX,4C00H    ;Exit to DOS
        INT     21H
BEGIN  ENDP
;-----
B10INIT PROC    NEAR
        MOV     AX,00H      ;Initialize mouse
        INT     33H
        CMP     AX,00       ;Mouse installed?
        JE      B90        ; no -- exit
        MOV     AX,01H      ;Show pointer
        INT     33H
B90:    RET          ;Return to caller
B10INIT ENDP
;-----
.286
D10PTR PROC    NEAR
D20:    MOV     AX,03H      ;Get pointer location
        INT     33H
        CMP     BX,01      ;Right button pressed?
        JE      D90        ; yes -- means exit
        SHR     CX,03      ;Divide pixel value
        SHR     DX,03      ; by 8
        CMP     CX,XBINARY ;Has pointer location
        JNE     D30        ; changed?
        CMP     DX,YBINARY ;
        JE      D20        ;no -- repeat operation
D30:    ; yes --
        MOV     XBINARY,CX ; save new locations
        MOV     YBINARY,DX ;
D90:    RET          ;Return to caller
D10PTR ENDP

```

```
G10CONV PROC    NEAR        ;AX = binary X or Y
                MOV        ASCVAL,2020H ;Clear ASCII field
                MOV        CX,10       ;Set divide factor
                LEA        SI,ASCVAL+1 ;Load ASCVAL address
                CMP        AX,CX       ;Compare location to 10
                JB         G30         ; lower -- bypass
                DIV        CL          ; higher -- div by 10
                OR         AH,30H      ;Insert ASCII 3s
                MOV        [SI],AH    ;Store in rightmost byte
                DEC        SI          ;Decr address of ASCVAL
G30:
                OR         AL,30H      ;Insert ASCII 3s
                MOV        [SI],AL    ;Store in leftmost byte
                RET          ;Return to caller
G10CONV ENDP
;-----
H10HIDE PROC    NEAR
                MOV        AX,02H     ;Hide pointer
                INT        33H
                RET          ;Return to caller
H10HIDE ENDP

Q10CLEAR        PROC    NEAR
                MOV        AX,0600H   ;Request clear screen
                MOV        BH,30H     ;Colors
                MOV        CX,00      ;Full
                MOV        DX,184FH   ; screen
                INT        10H
                RET          ;Return to caller
Q10CLEAR        ENDP

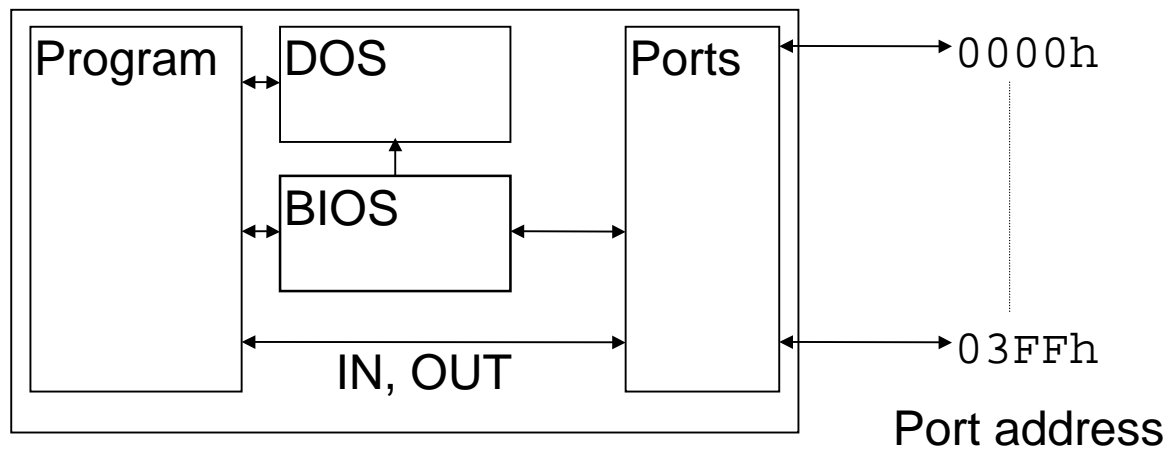
Q20CURS PROC    NEAR
                MOV        AH,02H     ;Set cursor
                MOV        BH,0       ;Page 0
                MOV        DH,0       ;Row
                MOV        DL,25      ;Column
                INT        10H
```

```
        RET          ;Return to caller
Q20CURS ENDP

Q30DISP PROC      NEAR
        MOV         AH,40H      ;Request display
        MOV         BX,01       ;Screen
        MOV         CX,14       ;Number of characters
        LEA         DX,DISPDATA ;Display area
        INT         21H
        RET          ;Return to caller
Q30DISP ENDP
        END         BEGIN
```

## Communicating Directly through PORTS

You can bypass DOS and the BIOS and go directly to the address through which there is a connection to the outside world (but...beware!)



Major port addresses:

060h	Input from keyboard
061h	Speaker
200h-20Fh	Game controller
3BCh-3BFh	LPT1: (parallel port)
3F8h-3FFh	COM1: (serial port)

On power up, the BIOS places the port addresses for serial and parallel ports that it finds in a BIOS table, starting at location 40[0]:00.

For example, LPT1 is put at 40:08, and LPT2 at 40:0A

```
TITLE    P21PORT Switch printer ports LPT1 & 2
BIOSDAT SEGMENT AT 40H      ;BIOS data area
          ORG      8H        ;Printer port addr
PARLPRT DW      4 DUP(?)    ;4 words
BIOSDAT ENDS

CODESG   SEGMENT PARA 'code'
          ASSUME   DS:BIOSDAT,CS:CODESG
          ORG      100H
BEGIN:
          MOV      AX,BIOSDAT
          MOV      DS,AX

          MOV      AX,PARLPRT(0) ;LPT1 address to AX
          MOV      BX,PARLPRT(2) ;LPT2 address to BX
          MOV      PARLPRT(0),BX ;Exchange addresses
          MOV      PARLPRT(2),AX ;Exchange addresses
          MOV      AX,4C00H      ;Exit to DOS
          INT      21H
CODESG   ENDS
          END      BEGIN
```

Exchange ports LPT1: and LPT2: by changing the BIOS table.

The BIOS segment is based at 40h.



```

TITLE    P21SOUND Produce sound from speaker
SOUNSG   SEGMENT PARA 'Code'
          ASSUME  CS:SOUNSG,DS:SOUNSG,SS:SOUNSG
          ORG    100H
BEGIN:   JMP     SHORT MAIN
; -----
DURTION  DW      1000           ;Length of tone
TONE     DW      256H          ;Frequency
; -----
MAIN     PROC     NEAR
          IN      AL,61H        ;Get port data
          PUSH   AX             ; and save
          CLI     ;Clear interrupts
          CALL   B10SPKR        ;Produce sound
          POP    AX             ;Reset
          OUT    61H,AL         ; port value
          STI     ;Reset interrupts
          RET
MAIN     ENDP

B10SPKR  PROC     NEAR
B20:     MOV     DX,DURTION     ;Set duration of sound
B30:
          AND    AL,11111100B   ;Clear bits 0 & 1
          OUT    61H,AL         ;Transmit to speaker
          MOV    CX,TONE        ;Set length
B40:
          LOOP   B40            ;Time delay
          OR     AL,00000010B   ;Set bit 1 on
          OUT    61H,AL         ;Transmit to speaker
          MOV    CX,TONE        ;Set length
B50:
          LOOP   B50            ;Time delay
          DEC    DX             ;Reduce duration
          JNZ   B30             ;Continue?
          SHL   DURTION,1       ; no - incr length
          SHR   TONE,1          ;Reduce frequency
          JNZ   B20             ;Now zero?

```

```
        RET      ;  yes - return
B10SPKR ENDP

SOUNSG ENDS
        END      BEGIN
```

## Exercises – Lecture 23

1. Sketch out the code (it is too long to write the complete code) describing how you would use the mouse to select items from a menu (text mode).